



计算机科学

COMPUTER SCIENCE

移动边缘计算中基于Stackelberg模型的分布式定价与计算卸载

陈序展, 林兵, 陈星

引用本文

陈序展, 林兵, 陈星. 移动边缘计算中基于Stackelberg模型的分布式定价与计算卸载[J]. 计算机科学, 2023, 50(7): 278-285.

CHEN Xuzhan, LIN Bing, CHEN Xing. [Stackelberg Model Based Distributed Pricing and Computation Offloading in Mobile Edge Computing](#) [J]. Computer Science, 2023, 50(7): 278-285.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[中继选择和队列稳定动态能量优化策略](#)

Dynamic Energy Optimization Strategy Based on Relay Selection and Queue Stability
计算机科学, 2023, 50(6A): 220100082-8. <https://doi.org/10.11896/jsjcx.220100082>

[基于线性规划松弛的移动边缘计算卸载模型](#)

MEC Offloading Model Based on Linear Programming Relaxation
计算机科学, 2023, 50(6A): 211200229-5. <https://doi.org/10.11896/jsjcx.211200229>

[面向能耗优化和负载均衡的边缘服务器放置研究](#)

Edge Server Placement for Energy Consumption and Load Balancing
计算机科学, 2023, 50(6A): 220300088-5. <https://doi.org/10.11896/jsjcx.220300088>

[空-天-地一体化移动边缘计算系统的部署优化和计算卸载](#)

Deployment Optimization and Computing Offloading of Space-Air-Ground Integrated Mobile Edge Computing System
计算机科学, 2023, 50(2): 69-79. <https://doi.org/10.11896/jsjcx.220600057>

[一种基于博弈论的移动边缘计算资源分配策略](#)

Resource Allocation Strategy Based on Game Theory in Mobile Edge Computing
计算机科学, 2023, 50(2): 32-41. <https://doi.org/10.11896/jsjcx.220300198>

移动边缘计算中基于 Stackelberg 模型的分布式定价与计算卸载

陈序展^{1,2} 林兵^{2,3} 陈星^{1,2}

1 福州大学计算机与大数据学院 福州 350108

2 福建省网络计算与智能信息处理重点实验室 福州 350108

3 福建师范大学物理与能源学院 福州 350117

(chenxuzhan1998@163.com)

摘要 移动边缘计算(Mobile Edge Computing, MEC)作为一种新颖的计算范式,通过将计算任务从移动设备卸载到物理上临近的网络边缘,为移动设备提供低延迟和灵活的计算与通信服务。然而,由于边缘服务器和移动设备通常属于不同的主体,它们之间的利益冲突给 MEC 系统带来了很大的挑战。因此,为多边缘服务器多移动设备的 MEC 系统设计一种定价和计算卸载方案,最大化边缘服务器的效用并优化移动设备的体验质量至关重要。考虑到多边缘服务器多移动设备之间复杂的相互作用,采用多领导者多追随者 Stackelberg 模型来分析它们之间的相互作用。其中,边缘服务器作为领导者为其计算资源设定价格,移动设备作为追随者根据边缘服务器的定价来调整卸载策略。在 Stackelberg 模型的基础上,提出了一种基于次梯度法的分布式迭代算法,该算法可以有效收敛到 Stackelberg 均衡。仿真实验结果表明,所提方案能够在提高边缘服务器效用的同时保证移动设备的体验质量。

关键词: 移动边缘计算; Stackelberg 模型; 多领导者多追随者; 定价; 计算卸载

中图分类号 TP393

Stackelberg Model Based Distributed Pricing and Computation Offloading in Mobile Edge Computing

CHEN Xuzhan^{1,2}, LIN Bing^{2,3} and CHEN Xing^{1,2}

1 College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China

2 Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing, Fuzhou 350108, China

3 College of Physics and Energy, Fujian Normal University, Fuzhou 350117, China

Abstract As a novel computing paradigm, mobile edge computing(MEC) provides low latency and flexible computing and communication services for mobile devices by offloading computing tasks from mobile devices to the physically proximal network edge. However, because edge servers and mobile devices often belong to different parties, the conflicts of interest between them present a great challenge for MEC systems. Therefore, it is important to design a pricing and computing offloading scheme for MEC systems with multiple edge servers and mobile devices to maximize the utility of edge servers and optimize the quality of experience for mobile devices. Considering the complex interaction between multi-edge servers and mobile devices, the multi-leader and multi-follower Stackelberg model is used to analyze the interaction between them. The edge server acts as the leader to set the price for its computing resources, and the mobile device as the follower adjusts the offloading strategy according to the pricing of the edge server. Based on the Stackelberg model, a distributed iterative algorithm based on subgradient method is proposed, which can effectively converge to Stackelberg equilibrium. Simulation results show that the proposed scheme can improve the utility of edge server and guarantee the experience quality of mobile devices.

Keywords Mobile edge computing, Stackelberg model, Multi-leader multi-followers, Pricing, Computation offloading

到稿日期:2022-05-26 返修日期:2022-10-10

基金项目:国家自然科学基金(62072108);福建省自然科学基金杰青项目(2020J06014);国家重点研发计划(2017YFB1002000);福建省高校产学研合作项目(2022H6024)

This work was supported by the National Natural Science Foundation of China(62072108), Natural Science Foundation of Fujian Province for Distinguished Young Scholars(2020J06014), National Key R & D Program of China(2017YFB1002000) and University-Industry Cooperation of Fujian Province(2022H6024).

通信作者:林兵(WheelLX@163.com)

1 引言

随着物联网技术的快速发展和第五代移动通信技术(5G)的普及,越来越多新型的智能移动应用开始进入人们的日常生活中,如增强现实(Augmented Reality, AR)、虚拟现实(Virtual Reality, VR)、智能交通等^[1]。这些新型的智能移动应用基本上都是计算密集型应用,通常涉及复杂的数据分析和处理,总是需要较大的计算能力和较高的能量消耗^[2]。然而,轻量级移动设备的计算能力和电池容量有限,不能够很好地支持这些新型的智能移动应用,移动设备在本地处理所有原始数据,难以满足新型移动应用的实时计算需求^[3]。

移动云计算(Mobile Cloud Computing, MCC)作为移动计算和云计算的结合,提供了一种可行的解决方案。移动设备可以将其任务卸载到远程数据中心,以利用强大的中央处理器(CPU)和云的充足存储能力。然而,由于 MCC 所依赖的数据中心在地理位置上远离移动设备,因此会导致相当大的传输延迟,不能满足 5G 和物联网的严苛要求(如低时延等),影响了移动设备的体验质量^[4]。

移动边缘计算(MEC),作为一种新颖的移动云计算范式,可以有效地解决 MCC 存在的问题^[5]。MEC 使移动设备能够通过无线网络(如 WiFi 和 5G)将任务卸载到边缘服务器^[6]。由于边缘服务器具有相对丰富的计算资源,并且更加接近移动设备,MEC 可以提供低延迟、低能耗、高质量的计算服务,从而提高移动设备的寿命和任务处理能力。

尽管 MEC 潜力巨大,但它仍处于起步阶段,并且面临着诸多挑战。如在现实的 MEC 系统中,边缘服务器和移动设备通常属于不同的主体,彼此之间存在着利益冲突以及边缘服务器通常只具有有限的计算能力等。因此,如何合理地边缘服务器的计算资源进行定价,以及移动设备如何有效地将任务卸载到边缘服务器,以通过高效的计算卸载策略充分挖掘 MEC 系统的潜力,已经成为研究的热点^[7-8]。

在多边缘服务器多移动设备场景下,移动设备的卸载策略和边缘服务器的定价策略彼此之间不再是相互独立的,而是会相互影响。为了描述这种动态的博弈,边缘服务器和移动设备之间的交互可以被建模为 Stackelberg 博弈^[9]。目前已经有解决 Stackelberg 博弈问题的方法^[10-11],但是这些方法都是集中式的。换句话说,这些解决方案都假设边缘服务器不仅可以知道与 Stackelberg 博弈相关的所有信息(例如每个移动设备的效用和每个边缘服务器的效用),而且有权调整所有边缘服务器的定价策略和所有移动设备的计算卸载策略。然而,在现实的 MEC 系统中,边缘服务器和移动设备属于不同的主体,以效用为导向,只关注自己的效用。因此,需要一种可行且有效的分布式定价和计算卸载方案来解决它们之间的利益冲突,使得边缘服务器在向移动设备提供计算资源的同时可以获得一定的回报,移动设备则可以在为计算资源付费的同时获得更好的用户体验。

鉴于现有文献的局限性,我们提出了一种在多边缘服务器多移动设备场景下的分布式定价与计算卸载方案。边缘服务器与移动设备之间的互动被建模为一个多领导者多追随者的双层 Stackelberg 博弈。我们设计了基于次梯度法的分布

式迭代算法来解决该博弈问题。本文的主要贡献包括:

(1)将多边缘服务器多移动设备场景下的边缘服务器资源定价和移动设备计算卸载问题,建模为一个多领导者多追随者 Stackelberg 博弈,其中边缘服务器作为领导者,移动设备作为追随者。

(2)提出了一种基于次梯度法的分布式迭代算法,可以有效收敛到 Stackelberg 均衡。

(3)设计了完备合理的仿真环境并进行了大量的仿真实验,以评估所提出的分布式定价和计算卸载方案的性能。实验结果表明,本文方案可以在保证移动设备体验质量的同时大幅提高边缘服务器的效用。

本文第 2 节回顾了相关工作;第 3 节描述了系统模型;第 4 节在系统模型的基础上分析了边缘服务器和移动设备之间的 Stackelberg 博弈;第 5 节基于逆向归纳法对博弈参与者的最优策略进行了分析;第 6 节进行了仿真实验评估与分析;最后总结全文。

2 相关工作

作为一种新兴的计算架构,近年来,MEC 及其相关技术的研究受到了学术界和工业界的广泛关注^[12]。通常,移动设备受限于存储空间、电池寿命和计算能力^[13],而 MEC 则可以在靠近移动设备的网络边缘提供计算能力,以帮助资源有限的移动设备完成计算密集型和延迟敏感型任务^[14]。本节回顾并分析了 MEC 中计算卸载问题的相关工作。

考虑不同的因素或采用不同的方法,许多学者对 MEC 中的计算卸载问题展开了研究。Chen 等^[15]提出了一种基于遗传算法算子的自适应粒子群优化算法的节能卸载方法。仿真结果表明,与其他经典方法相比,该方法能显著降低能耗。Lyu 等^[16]针对无人机使能的 MEC 系统,利用块坐标下降算法,联合优化带宽分配、任务卸载时间分配和无人机轨迹来最大化整个系统的计算量。Guo 等^[17]提出并设计了一种新的基于李亚普诺夫优化的部分计算卸载算法,以求解由单个边缘服务器和多个用户组成的 MEC 系统的数据分区计算任务卸载问题。Wang 等^[18]针对物联网中的软件定义移动边缘计算(SD-MEC),提出了一种基于分布式深度学习的计算卸载和资源分配(DDL-CORA)算法,以最小化分布式密集物联网(IoT)环境中加权延迟和功耗的效用。Zhou 等^[19]针对动态多用户 MEC 系统中计算卸载和资源分配的联合优化问题,提出了一种基于双深度 Q 网络的方法,来最小化整个 MEC 系统的能量消耗。Nie 等^[20]针对无人机使能的 MEC 系统,提出了一种基于联邦学习的深度强化学习算法,仿真结果表明,通过使用该算法,用户终端可以以节能的方式将计算敏感型任务卸载到最优的无人机上。Lu 等^[21]针对多基站多用户终端的密集网络中可划分应用的计算卸载问题,提出了一种进化博弈方法,用于最小化任务的完成时间。

上述研究主要关注用户层面的计算卸载问题,这些研究假定边缘服务器免费提供计算资源,忽略了边缘服务器定价策略对卸载决策的影响。实际上,边缘服务器的部署和运行都需要一定的成本,边缘服务器对其计算资源进行定价和销售更加符合实际的情况。

Sun 等^[22]基于拍卖理论,提出了一种基于盈亏平衡的双向拍卖方案和一种更有效的基于动态定价的双向拍卖方案,以实现高系统效率的定价机制。然而,他们仅仅关注边缘服务器方面的资源分配,忽略了移动设备卸载策略的影响。Du 等^[23]针对 MEC 使能的区块链系统,提出了一种基于 A3C 深度强化学习算法的低复杂度算法,用于联合优化通信和计算资源定价与分配。然而,他们没有考虑同时优化边缘服务器和移动设备的目标。Yan 等^[24]提出了一种 MEC 服务定价方案,通过一个低复杂度的算法来联合优化基站的定价和服务缓存,以协调服务缓存决策并控制蜂窝网络中无线设备的任务卸载行为。然而,他们只考虑了 MEC 环境中只有一个边缘服务器的情况。Xu 等^[25]提出了一种基于 Stackelberg 博弈的资源定价和交易方案,在边缘服务器和无人机之间分配边缘计算资源。然而,他们没有考虑边缘服务器之间的竞争。

与上述工作不同,本文将研究重点放在多边缘服务器多移动设备场景下的定价和计算卸载问题上。在多边缘服务器多移动设备场景下,多个边缘服务器之间存在竞争,每个移动设备可以按比例将其计算任务卸载给多个边缘服务器,这无疑增加了分析和解决定价和计算卸载问题的难度。因此,对于多边缘服务器多移动设备场景,以完全分布式的方式同时解决定价和计算卸载问题仍然是一个十分具有挑战性的问题。

3 系统模型

如图 1 所示,我们考虑一个由 m 个边缘服务器和 n 个移动设备组成的 MEC 系统。 m 个边缘服务器表示为 $\mathbf{M} = \{1, 2, \dots, m\}$, n 个移动设备表示为 $\mathbf{N} = \{1, 2, \dots, n\}$ 。移动设备的计算任务可以按比特任意划分,用于部分本地计算和部分卸载^[26]。

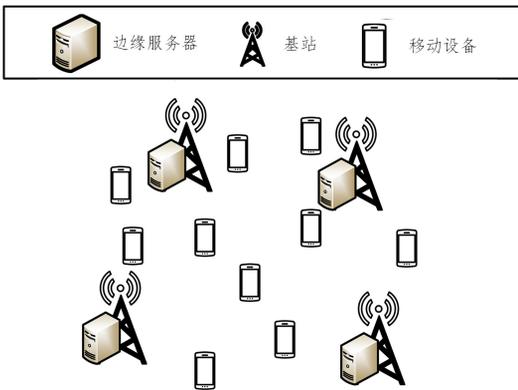


图 1 多边缘服务器多移动设备的 MEC 系统

Fig. 1 Multi-edge server multi-mobile device MEC system

本文假定移动设备 $i, i \in \mathbf{N}$, 有一项待执行的计算任务, 记为 $Task_i = (D_i, S_i)$, 其中 D_i 表示该任务输入数据量的大小(以比特为单位), S_i 表示该任务每比特的数据量所需的计算资源量(以 CPU 的计算周期数为单位)。假设这些任务是计算密集型的、相互独立的, 可以按比特任意划分的, 并且可以在移动设备中本地执行, 或者通过计算卸载在边缘服务器上远程执行^[27]。移动设备 i 以一定的比例 $x_{i,j}$ 将其计算任务卸载到边缘服务器 $j, j \in \mathbf{M}, 0 \leq x_{i,j} \leq 1$ 。因此, 对于移动设备 i , 其卸载策略向量表示为:

$$\mathbf{x}_i = (x_{i,1}, \dots, x_{i,m})^T \quad (1)$$

其中, $0 \leq \sum_{j \in \mathbf{M}} x_{i,j} \leq 1$ 。

本文中涉及的主要符号及其定义如表 1 所列。

表 1 主要符号及其定义

Table 1 Principal symbols and their definitions

符号	定义
m	边缘服务器的数量
n	移动设备的数量
\mathbf{M}	边缘服务器的集合
\mathbf{N}	移动设备的集合
$Task_i$	移动设备 i 的计算任务
D_i	移动设备 i 的计算任务的数据量
S_i	移动设备 i 的计算任务每比特需要的计算资源量
$x_{i,j}$	移动设备 i 的计算任务卸载到边缘服务器 j 的比例
f_i^{MD}	移动设备 i 的 CPU 计算频率
B	信道带宽
P_i	移动设备 i 的传输功率
$h_{i,j}$	移动设备 i 和边缘服务器 j 间的信道增益
σ^2	高斯白噪声功率
$r_{i,j}$	移动设备 i 和边缘服务器 j 间的传输速率
f_j^{Edge}	边缘服务器 j 的 CPU 计算频率
v_i	加权因子
p_j	边缘服务器 j 的单位计算资源定价
c	边缘服务器单位计算资源的能耗成本
C_{fixed}	边缘服务器启动的固定成本

3.1 本地计算

对于移动设备 i , 由于有 $\sum_{j \in \mathbf{M}} x_{i,j}$ 比例的计算任务卸载到了边缘服务器上, 因此剩下 $1 - \sum_{j \in \mathbf{M}} x_{i,j}$ 比例的计算任务在本地执行。根据 $Task_i$ 的定义, 执行该任务所需的总计算周期数为 $D_i \times S_i$ 。因此, 在移动设备 i 上执行计算任务所需的时间为:

$$T_{i,loc}^{loc} = \frac{(1 - \sum_{j \in \mathbf{M}} x_{i,j}) D_i S_i}{f_i^{MD}} \quad (2)$$

其中, f_i^{MD} 表示移动设备 i 的 CPU 计算频率, 它由每秒的 CPU 周期数来度量。

3.2 计算卸载

当移动设备 i 的部分计算任务卸载到边缘服务器 j 时, 其卸载的计算任务的完成时间由 3 个部分构成, 包括数据从移动设备 i 上传到边缘服务器 j 所需的上行传输时间 $T_{i,j}^{tran}$ 、边缘服务器 j 执行计算任务所需的时间 $T_{i,j}^{edge}$ 以及计算结果从边缘服务器 j 回传到移动设备 i 的下行传输时间 $T_{i,j}^{back}$ 。

计算任务首先需要被传输到边缘服务器。我们忽略了计算卸载时移动设备对信道造成的干扰, 根据香农公式^[28], 移动设备 i 与边缘服务器 j 之间的数据传输速率为:

$$r_{i,j} = B \log_2 \left(1 + \frac{P_i h_{i,j}}{\sigma^2} \right) \quad (3)$$

其中, B 表示信道带宽, P_i 表示移动设备 i 的传输功率, $h_{i,j}$ 表示移动设备 i 和边缘服务器 j 之间的信道增益, σ^2 表示通信链路中的高斯白噪声功率。

因此移动设备 i 向边缘服务器 j 传输数据所需的时间为:

$$T_{i,j}^{tran} = \frac{x_{i,j} D_i}{r_{i,j}} \quad (4)$$

计算任务在边缘服务器 j 上的执行时间为:

$$T_{i,j}^{edge} = \frac{x_{i,j} D_i S_i}{f_j^{Edge}} \quad (5)$$

其中, f_j^{Edge} 表示边缘服务器 j 的 CPU 计算频率, 它由每秒的 CPU 周期数来度量。

当计算任务执行完成后, 其执行结果需要被回传到相应的移动设备上。但是, 由于下行链路速率通常远高于上行链路速率, 并且输出结果的数据量大小通常远小于任务输入的数据量大小, 因此结果回传的下行传输时间 $T_{i,j}^{\text{back}}$ 可以忽略不计^[29-30]。

综上所述, 当移动设备 i 将部分计算任务卸载到边缘服务器 j 上执行时, 任务的完成时间可以表示为:

$$T_{i,j}^{\text{off}} = T_{i,j}^{\text{tran}} + T_{i,j}^{\text{edge}} \quad (6)$$

由于每个计算任务可以被分成多个部分, 在本地和各个边缘服务器上同时并行计算^[31]。因此, 任务延迟由多个并行执行部分的最大值决定。移动设备 i 的计算任务的最大延迟为:

$$T_i^{\text{max}} = \max\{T_{i,1}^{\text{off}}, T_{i,2}^{\text{off}}, \dots, T_{i,m}^{\text{off}}, T_i^{\text{loc}}\} \quad (7)$$

4 基于 Stackelberg 博弈的问题形式化

本节利用多领导者多追随者 Stackelberg 博弈模型对 MEC 系统进行建模。Stackelberg 博弈模型是一种层次结构清晰的博弈模型, 该模型包括 4 个要素: 领导者、追随者、策略集和效用函数。如图 2 所示, 在本文模型中, 边缘服务器是领导者, 移动设备是追随者。边缘服务器的策略是其对自身单位 CPU 计算资源的定价 p , 移动设备的策略是其卸载策略 x 。不同于以往的计算卸载方案仅仅通过最小化时延来考虑移动设备的效用^[32], 本文方案分别为移动设备和边缘服务器制定了效用。接下来, 我们将分别分析和阐述移动设备和边缘服务器的效用函数和优化问题。

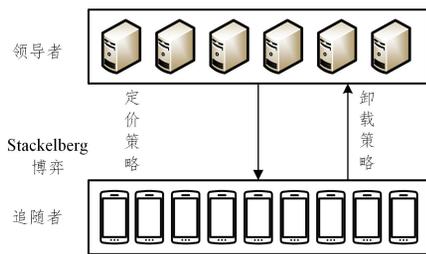


图 2 Stackelberg 博弈模型

Fig. 2 Stackelberg game model

4.1 移动设备的问题形式化

移动设备的体验质量(QoE)会受到任务执行延迟和支付成本的影响。为了实现低延迟, 移动设备倾向于将更多数据卸载到边缘服务器。然而, 如果移动设备卸载太多数据, 其支付成本会很高。因此, 需要权衡任务执行延迟和支付成本。综上所述, 我们将移动设备的体验质量量化为一个由时间延迟成本和支付成本组成的负效用函数:

$$U_i(x_i) = v_i T_i^{\text{max}} + \sum_{j \in \mathbf{M}} p_j x_{i,j} D_i S_i \quad (8)$$

其中, v_i 是时间延迟成本相对于支付成本的加权因子, 用于统一时间延迟成本和支付成本的维度, 以便可以对它们求和^[33]; p_j 表示边缘服务器 j 对其自身单位 CPU 计算周期计算资源的定价; $\sum_{j \in \mathbf{M}} p_j x_{i,j} D_i S_i$ 表示移动设备 i 为其卸载到各个边缘服务器的计算任务所支付的总成本。

移动设备的目标是最小化其负效用。因此, 根据式(8),

移动设备需要减少其最大时延, 并减少其支付给边缘服务器的成本。在数学上, 移动设备 i 的优化问题 P1 可以被定义为:

$$\begin{aligned} \text{P1: } & \min U_i(x_i) \\ \text{s. t. } & \text{C1: } 0 \leq x_{i,j} \leq 1, \forall j \in \mathbf{M} \\ & \text{C2: } \sum_{i \in \mathbf{N}} x_{i,j} D_i S_i \leq F_j, \forall j \in \mathbf{M} \\ & \text{C3: } 0 \leq \sum_{j \in \mathbf{M}} x_{i,j} \leq 1 \end{aligned} \quad (9)$$

其中, F_j 表示边缘服务器 j 的 CPU 周期资源上限。第一个约束 C1 表示移动设备 i 的计算任务卸载到边缘服务器 j 的比例必须在 0 到 1 之间, 第二个约束 C2 保证不同移动设备卸载到同一边缘服务器的 CPU 周期总数不能超过其 CPU 周期资源上限, 第三个约束 C3 保证移动设备卸载出去的计算任务不会超过其需要处理的计算任务。

4.2 边缘服务器的问题形式化

边缘服务器通过将其计算资源出售给移动设备而获取收益, 同时, 边缘服务器的部署和运行都需要一定的成本。因此, 我们将边缘服务器的效用定义为边缘服务器出售计算资源而获得的收入减去边缘服务器运行的能耗成本与边缘服务器的固定成本。边缘服务器 j 的效用函数为:

$$\Pi_j(p_j) = \sum_{i \in \mathbf{N}} p_j x_{i,j} D_i S_i - \sum_{i \in \mathbf{N}} c x_{i,j} D_i S_i - C_{\text{fixed}} \quad (10)$$

其中, c 表示边缘服务器每单位 CPU 周期计算资源的能耗成本(例如电费), C_{fixed} 表示边缘服务器启动的固定成本。

边缘服务器的目标是最大化其自身的效用。在多边缘服务器多移动设备场景下, 由于每个移动设备都会按照最优的卸载策略来卸载其任务, 因此边缘服务器之间将存在竞争。如果边缘服务器降低其定价与其他边缘服务器进行竞争, 移动设备卸载到该边缘服务器的任务量会变大。然而, 由于设定的价格低, 边缘服务器从每个单位计算资源获得的收入将减少。类似地, 当边缘服务器提高其定价时, 即使边缘服务器从单位计算资源获得了更高的收入, 由于其他边缘服务器单位计算资源的价格更低, 卸载到该边缘服务器的计算资源量会减少。因此, 每个边缘服务器都需要对其定价进行权衡, 以找到一个合适的定价来使其效用最大化。在数学上, 边缘服务器 j 的优化问题 P2 可以表示为:

$$\begin{aligned} \text{P2: } & \max \Pi_j(p_j) \\ \text{s. t. } & \text{C1: } p_j \geq c \end{aligned} \quad (11)$$

其中, 约束 C1 表示边缘服务器对其单位 CPU 周期计算资源的定价不能低于其单位 CPU 周期计算资源的能耗成本。

4.3 多领导者多追随者的 Stackelberg 博弈

问题 P1 和问题 P2 共同构成了一个多领导者多追随者 Stackelberg 博弈。其中, 边缘服务器作为领导者, 移动设备作为追随者。博弈的参与者集合由集合 \mathbf{M} 和 \mathbf{N} 组成。博弈的目标是找到领导和追随者都没有动机偏离的 Stackelberg 博弈均衡解决方案。我们将 Stackelberg 博弈均衡定义如下。

定义 1(Stackelberg 均衡, SE) 用 $((\mathbf{P}, \mathbf{X}), (\Pi, U))$ 表示具有 m 个领导者和 n 个追随者的多领导者多追随者 Stackelberg 博弈。 $\mathbf{P} = \mathbf{P}_1 \times \mathbf{P}_2 \times \dots \times \mathbf{P}_m$ 和 $\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \dots \times \mathbf{X}_n$ 分别表示领导和追随者的策略集合; Π 是领导者的效用

函数; U 是追随者的负效用函数; \mathbf{p}_j 表示领导者 j 的策略集合, \mathbf{p}_{-j} 表示除领导者 j 以外所有领导者的策略集合, \mathbf{x}_i 表示追随者 i 的策略集合, \mathbf{x}_{-i} 表示除追随者 i 以外所有追随者的策略集合。一组策略集合 $\mathbf{p}^* \in \mathbf{P}$ 和 $\mathbf{x}^* \in \mathbf{X}$ 是多领导者多追随者 Stackelberg 博弈的均衡, 如果对于任意的领导者 j 和追随者 i 满足:

$$\prod_j(\mathbf{p}_j^*, \mathbf{p}_{-j}^*, \mathbf{x}^*) \geq \prod_j(\mathbf{p}_j, \mathbf{p}_{-j}^*, \mathbf{x}^*) \quad (12)$$

和

$$U_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*, \mathbf{p}^*) \leq U_i(\mathbf{x}_i, \mathbf{x}_{-i}^*, \mathbf{p}^*) \quad (13)$$

5 博弈分析

Stackelberg 博弈问题可以用逆向归纳法来解决, 即首先解决追随者的 P1 问题, 然后解决领导者的 P2 问题。本节首先分析作为追随者的移动设备的卸载策略, 然后分析作为领导者的边缘服务器的定价策略。

5.1 移动设备的卸载策略

在给边缘服务器的定价的情况下, 移动设备根据各个边缘服务器的定价做出自己的最优卸载决策。移动设备的决策问题如下:

$$\begin{aligned} \text{P1: } \min U_i(x_i) &= v_i T_i^{\max} + \sum_{j \in \mathbf{M}} p_j x_{i,j} D_i S_i \\ \text{s. t. C1: } 0 &\leq x_{i,j} \leq 1, \forall j \in \mathbf{M} \\ \text{C2: } \sum_{i \in \mathbf{N}} x_{i,j} D_i S_i &\leq F_j, \forall j \in \mathbf{M} \\ \text{C3: } 0 &\leq \sum_{j \in \mathbf{M}} x_{i,j} \leq 1 \end{aligned} \quad (14)$$

引理 1 问题 P1 是一个凸优化问题。

证明: 显然, 问题 P1 的可行域是非空凸集。因此, 我们只需要证明问题 P1 的目标函数是凸的。 $T_i^{\max} = \max\{T_{i,1}^{\text{off}}, T_{i,2}^{\text{off}}, \dots, T_{i,m}^{\text{off}}, T_i^{\text{loc}}\}$ 中各子项均为凸函数, 故 T_i^{\max} 是逐点最大值函数, 根据逐点最大值函数的保凸性, 可得 T_i^{\max} 为凸函数, 因此 $v_i T_i^{\max}$ 为凸函数。同时, $\sum_{j \in \mathbf{M}} p_j x_{i,j} D_i S_i$ 也是一个关于变量 x_i 的凸函数。凸函数的非负加权也是凸函数, 因此问题 P1 是一个凸优化问题, 引理 1 为真。

从上述证明可以得出, 问题 P1 是一个凸优化问题。由于难以直接给出关于优化变量 x_i 的解析解, 因此我们借助 CVX 工具包来进行求解, 得到移动设备根据边缘服务器定价的卸载策略。

5.2 边缘服务器的定价策略

边缘服务器作为博弈的领导者, 会根据移动设备的卸载策略调整其定价, 使其效用函数最大化。在本问题中, 由于追随者策略的解析解难以直接求得, 领导者的效用函数是线性的, 因此不可能使用传统的基于梯度的优化方法来求解 Stackelberg 博弈。为了调节边缘服务器的行为并获得稳定且最优的解决方案, 本文提出了一种基于次梯度法的分布式迭代算法, 该算法能够有效收敛到 Stackelberg 博弈均衡, 即 SE。次梯度法在 Stackelberg 博弈中被经常应用于无法获得追随者解析解的博弈模型, 例如文献[34]。

在算法初始化时, 每个边缘服务器随机设定一个价格。在每一次迭代中, 边缘服务器以步长 δ 增加或减少其定价, 并预测获得的效用。如果定价增加 δ 时获得的效用最高, 那么

在下一轮迭代中定价将增加 δ 。如果定价降低 δ 时获得的效用最高, 那么在下一轮迭代中定价将降低 δ 。否则, 边缘服务器将保持当前定价不变。直到所有边缘服务器的定价相比上一轮迭代没有变化时, 算法停止。在有限的迭代次数内, 所有边缘服务器都能够确定其效用最高的最佳决策。我们假设 \mathbf{p} 是迭代中所有边缘服务器的定价策略, $\mathbf{p}_{\text{old}_j}$ 是上一次迭代中边缘服务器 j 的定价策略, $\mathbf{p}_{\text{old}_{-j}}$ 是上一次迭代中除边缘服务器 j 之外的所有边缘服务器的定价策略。详细的算法如算法 1 所示。然后, 对于所提算法, 引理 2 成立。

算法 1 基于次梯度法的分布式迭代算法

输入: $D_i, S_i, f_i^{\text{MD}}, r_{i,j}, f_j^{\text{Edge}}, v_i, c, C_{\text{fixed}}, F_j$

输出: $\mathbf{p}^*, \mathbf{x}^*$

1. Initially each edge server sets a random price
2. while at least one edge server adjust its price do
3. for mobile device i do
4. Based on the price set by all edge servers, each mobile device determines the optimal offloading strategy
5. end for
6. for edge server j do
7. Each edge server stores the current value of the price, $\mathbf{p}_{\text{old}} = \mathbf{p}$
8. Each edge server tries to increase or decrease its price with a small step δ , and calculates its own utility based on the prediction of the followers' optimal strategies
9. if $\Pi_j(\mathbf{p}_{\text{old}_j} + \delta, \mathbf{p}_{\text{old}_{-j}}) \geq \Pi_j(\mathbf{p}_{\text{old}_j}, \mathbf{p}_{\text{old}_{-j}})$ and $\Pi_j(\mathbf{p}_{\text{old}_j} + \delta, \mathbf{p}_{\text{old}_{-j}}) \geq \Pi_j(\mathbf{p}_{\text{old}_j} - \delta, \mathbf{p}_{\text{old}_{-j}})$ then
10. $\mathbf{p}_j = \mathbf{p}_{\text{old}_j} + \delta$
11. else
12. if $\Pi_j(\mathbf{p}_{\text{old}_j} - \delta, \mathbf{p}_{\text{old}_{-j}}) \geq \Pi_j(\mathbf{p}_{\text{old}_j}, \mathbf{p}_{\text{old}_{-j}})$ and $\Pi_j(\mathbf{p}_{\text{old}_j} - \delta, \mathbf{p}_{\text{old}_{-j}}) \geq \Pi_j(\mathbf{p}_{\text{old}_j} + \delta, \mathbf{p}_{\text{old}_{-j}})$ then
13. $\mathbf{p}_j = \mathbf{p}_{\text{old}_j} - \delta$
14. else
15. $\mathbf{p}_j = \mathbf{p}_{\text{old}_j}$
16. end if
17. end if
18. end for
19. end while

引理 2 当初始价格和步长 δ 固定时, 通过次梯度算法, 博弈总能收敛到唯一的结果, 即博弈的均衡。

证明: 在文献[35]和文献[36]中已经证明次梯度算法的收敛性。根据文献[35]和文献[36], 次梯度算法能够实现小范围的次优解。因此, 在给步长 δ 的情况下, 当次梯度算法收敛到次优解时, 每个边缘服务器不能单方面调整其定价以获得更高的效用。

当初始定价和步长 δ 固定时, 第二次迭代的结果也是固定的。根据数学归纳法, 我们假设在第 T 次迭代中, 边缘服务器的定价是固定的。然后在第 $T+1$ 次迭代中, 根据提出的次梯度算法, 步长 δ 是固定的, 并且从当前迭代到下一次迭代的方向是唯一的。因此, 在第 $T+1$ 次迭代中, 边缘服务器的定价也是固定的。基于上述分析, 当初始定价和步长 δ 固定时, 博弈可以收敛到一个唯一的结果。在边缘服务器之间达到均衡后, 移动设备总是用它们的最优策略来响应边缘

服务器的定价,因此这个结果必定是边缘服务器和移动设备都没有动力去偏离的 Stackelberg 博弈均衡解。

对于提出的基于次梯度法的分布式迭代算法,不需要所有的参与者在决策时严格同步。具体来说,对于每个移动设备,基于对来自所有边缘服务器的公布价格的观察,每个移动设备确定其最佳的卸载策略。在这种情况下,所有的移动设备不必在完全相同的时间做出决定。对于每个边缘服务器,基于对其他边缘服务器的公布价格的观察和对所有移动设备的当前行为的观察,它能够遵循本文算法来设置其价格,以提高其效用。换句话说,所有的边缘服务器也不需要同时决策,算法仍然可以收敛到同一个 Stackelberg 均衡。因此,该算法是一种完全分布式的算法。

6 实验评估

本节进行了大量的仿真实验来验证本文方案的性能。为了保证参数的真实性和可信度,本文参考了许多文献^[37-39]来设计实验仿真参数。仿真环境为 Python 3.8.5、Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz、8 GB 内存、Windows 10(64 位)。

6.1 实验设置

移动设备的计算任务量服从 100~500 kB 范围内的随机分布,以体现计算任务大小的差异。移动设备的计算任务每比特所需的计算资源量服从 500~1500 cycle 的随机分布,以体现执行任务所需的计算能力的差异。移动设备的本地计算频率服从 1.5~2.5 GHz 范围内的随机分布,以体现移动设备计算能力的差异。各个边缘服务器的 CPU 计算频率服从 3.5~4.5 GHz 范围内的随机分布,以体现边缘服务器的计算能力的差异。仿真参数的详细设置如表 2 所列。

表 2 系统参数

系统参数	值
D/kB	100~500
$S/(\text{cycles/bit})$	500~1500
f^{MD}/GHz	1.5~2.5
$f^{\text{Edge}}/\text{GHz}$	3.5~4.5
B/MHz	10
P/w	0.1
h/dBm	-50~-30
σ^2/dBm	-174
v	15~20
$F/(\text{cycles/slot})$	6×10^9
c	1×10^{-11}
C_{fixed}	1
δ	1×10^{-9}

6.2 实验结果与分析

图 3 给出了本文方案下所有移动设备的平均负效用与以下基线方案下所有移动设备的平均负效用。

- (1)本地卸载:每个移动设备在本地处理其所有任务。
- (2)平均卸载:每个移动设备以相同的比例将其任务卸载给每个边缘服务器。
- (3)PSO-GA:一种引入遗传算法来更新算子的改进粒子群优化算法。该算法继承了 PSO 算法易于实现、收敛快和精度高的特点,同时通过结合 GA 算法来更新算子,使得算法

能够跳出局部最优,获得更优质的解。

(4)最高数据速率卸载:每个移动设备贪婪地将其任务卸载到具有最小传输延迟的边缘服务器^[40]。

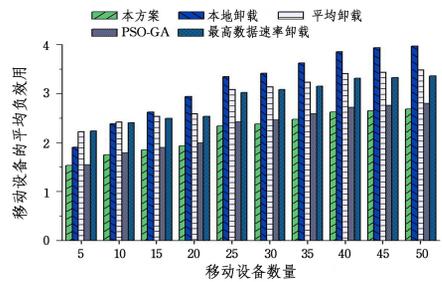


图 3 移动设备平均负效用的对比

Fig. 3 Comparison of average disutility of mobile devices

我们固定边缘服务器的数量为 3,改变移动设备的数量。可以看出,在不同的移动设备数量下,本文方案下移动设备的平均负效用均小于其他 4 种基线方案,即本文方案有效降低了移动设备执行计算任务的成本。每个移动设备在本地处理其所有任务的本地卸载方案虽然没有购买边缘服务器计算资源的支付成本,但由于计算任务全部在本地执行,其时延成本相对较大。每个移动设备以相同的比例将其任务卸载给每个边缘服务器的平均卸载方案虽然时延成本低,但购买边缘服务器计算资源的支付成本相对较大。而本文提出的方案对时延成本和支付成本进行了折中,实现了时延成本与支付成本之间的平衡。

图 4 给出了边缘服务器的平均效用随移动设备数量的变化趋势,我们固定边缘服务器的数量为 3。从图中可以看出,随着移动设备数量的增加,边缘服务器的平均效用不断增加。这是因为更多的移动设备将导致更多的卸载需求,从而导致边缘服务器平均效用的增加。此外,由于边缘服务器的计算资源有限,边缘服务器的平均效用的增长速度也在逐渐放缓。

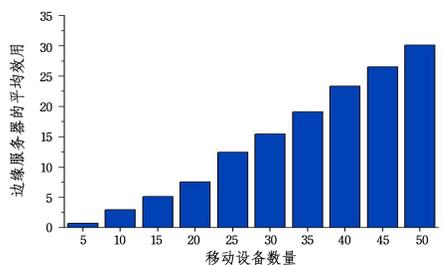


图 4 边缘服务器的平均效用随移动设备数量的变化趋势

Fig. 4 Trend of average utility of edge servers varies with the number of mobile devices

图 5 给出了边缘服务器的平均效用随边缘服务器数量的变化趋势。固定移动设备的数量为 30,边缘服务器的平均效用随着边缘服务器数量的增多而减少。这是因为更多的边缘服务器将增加边缘服务器之间的竞争并导致移动设备卸载量相对减少,从而导致边缘服务器平均效用降低。

图 6 给出了移动设备的平均负效用随边缘服务器数量的变化趋势。固定移动设备的数量为 30,移动设备的平均负效用随着边缘服务器数量的增多而减少。这是因为随着边缘服务器数量的增多,移动设备完成计算任务的成本也相对降低。

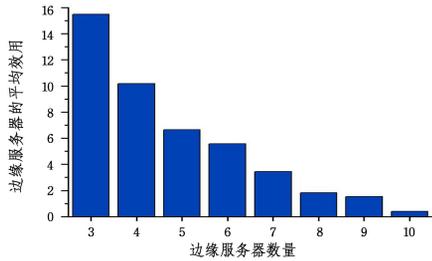


图5 边缘服务器的平均效用随边缘服务器数量的变化趋势

Fig. 5 Trend of average utility of edge servers varies with the number of edge servers

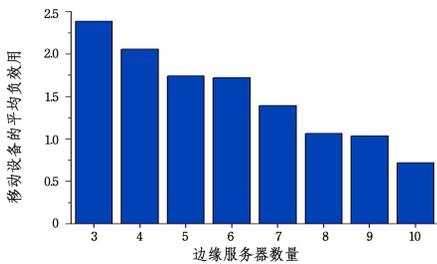


图6 移动设备的平均负效用随边缘服务器数量的变化趋势

Fig. 6 Trend of average disutility of mobile devices varies with the number of edge servers

图7给出了移动设备的平均负效用随移动设备数量的变化趋势。固定边缘服务器的数量为3,移动设备的平均负效用随着移动设备数量的增加而缓慢增加。结合图4可得,随着移动设备的数量从5台增加到50台,边缘服务器的平均效用增加了42倍,而移动设备的平均负效用仅增加了1.75倍,这说明本文方案可以在有效保证移动设备体验质量的同时,大幅度提高边缘服务器的效用。

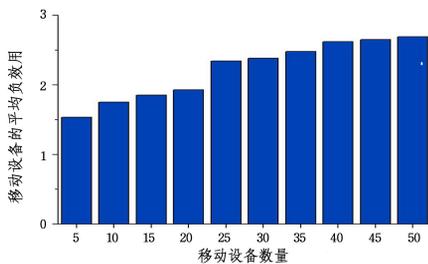


图7 移动设备的平均负效用随移动设备数量的变化趋势

Fig. 7 Trend of average disutility of mobile devices varies with the number of mobile devices

结束语 本文研究了多边缘服务器多移动设备场景下边缘服务器的资源定价和移动设备的计算卸载问题。边缘服务器与移动设备之间的交互被建模为一个多领导者多追随者Stackelberg博弈,其中边缘服务器作为领导者,移动设备作为追随者。本文提出了一种基于次梯度法的分布式迭代算法,用于有效地获得Stackelberg博弈的均衡解。实验结果表明,本文方案可以在显著提高边缘服务器效用的同时保证移动设备的体验质量。考虑到MEC系统的分布式环境可能导致边缘服务器与移动设备之间的交互存在安全性的问题,在未来的工作中,我们计划引入区块链技术来保证边缘服务器与移动设备之间资源交易的安全性和可追溯性。

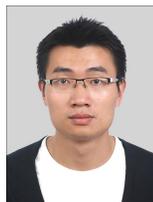
参考文献

- [1] LIU Y, PENG M, SHOU G, et al. Toward Edge Intelligence: Multiaccess Edge Computing for 5G and Internet of Things[J]. IEEE Internet of Things Journal, 2020, 7(8): 6722-6747.
- [2] GUO H, LIU J, QIN H. Collaborative Mobile Edge Computation Offloading for IoT over Fiber-Wireless Networks[J]. IEEE Network, 2018, 32(1): 66-71.
- [3] HU H, SONG W, WANG Q, et al. Energy Efficiency and Delay Tradeoff in an MEC-Enabled Mobile IoT Network[J]. IEEE Internet of Things Journal, 2022, 9(17): 15942-15956.
- [4] SHI W, CAO J, ZHANG Q, et al. Edge Computing: Vision and Challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.
- [5] MACH P, BECVAR Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J]. IEEE Communications Surveys Tutorials, 2017, 19(3): 1628-1656.
- [6] PAN J, MCELHANNON J. Future Edge Cloud and Edge Computing for Internet of Things Applications[J]. IEEE Internet of Things Journal, 2018, 5(1): 439-449.
- [7] SEO H, OH H, CHOI J K, et al. Differential Pricing-based Task Offloading for Delay-Sensitive IoT Applications in Mobile Edge Computing System[J]. IEEE Internet of Things Journal, 2022, 9(19): 19116-19131.
- [8] ZHANG K, GUI X, REN D, et al. Optimal pricing-based computation offloading and resource allocation for blockchain-enabled beyond 5G networks [J]. Computer Networks, 2022, 203: 108674.
- [9] LI Y, YANG B, WU H, et al. Joint Offloading Decision and Resource Allocation for Vehicular Fog-Edge Computing Networks: A Contract-Stackelberg Approach[J]. IEEE Internet of Things Journal, 2022, 9(17): 15969-15982.
- [10] LI F, YAO H, DU J, et al. Stackelberg Game-Based Computation Offloading in Social and Cognitive Industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics, 2020, 16(8): 5444-5455.
- [11] BAEK B, LEE J, PENG Y, et al. Three Dynamic Pricing Schemes for Resource Allocation of Edge Computing for IoT Environment[J]. IEEE Internet of Things Journal, 2020, 7(5): 4292-4303.
- [12] MAO Y, YOU C, ZHANG J, et al. A Survey on Mobile Edge Computing: The Communication Perspective[J]. IEEE Communications Surveys Tutorials, 2017, 19(4): 2322-2358.
- [13] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile Edge Computing: A Survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.
- [14] SPINELLI F, MANCUSO V. Toward Enabled Industrial Verticals in 5G: A Survey on MEC-Based Approaches to Provisioning and Flexibility[J]. IEEE Communications Surveys Tutorials, 2021, 23(1): 596-630.
- [15] CHEN X, ZHANG J, LIN B, et al. Energy-Efficient Offloading for DNN-Based Smart IoT Systems in Cloud-Edge Environments [J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(3): 683-697.

- [16] LYU L,ZENG F,XIAO Z, et al. Computation Bits Maximization in UAV-Enabled Mobile Edge Computing System[J]. IEEE Internet of Things Journal,2021,9(13):10640-10651.
- [17] GUO M,WANG W,HUANG X, et al. Lyapunov-Based Partial Computation Offloading for Multiple Mobile Devices Enabled by Harvested Energy in MEC[J]. IEEE Internet of Things Journal,2022,9(11):9025-9035.
- [18] WANG Z,LV T,CHANG Z. Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing[J]. Computer Networks,2022,205:108732.
- [19] ZHOU H,JIANG K,LIU X, et al. Deep Reinforcement Learning for Energy-Efficient Computation Offloading in Mobile-Edge Computing[J]. IEEE Internet of Things Journal,2022,9(2):1517-1530.
- [20] NIE Y,ZHAO J,GAO F, et al. Semi-Distributed Resource Management in UAV-Aided MEC Systems: A Multi-Agent Federated Reinforcement Learning Approach[J]. IEEE Transactions on Vehicular Technology,2021,70(12):13162-13173.
- [21] LU W,ZHANG X. Computation Offloading for Partitionable Applications in Dense Networks: An Evolutionary Game Approach[J]. IEEE Internet of Things Journal,2022,9(21):20985-20996.
- [22] SUN W,LIU J,YUE Y, et al. Double Auction-Based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics,2018,14(10):4692-4701.
- [23] DU J,CHENG W,LU G, 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.
- [24] YAN J,BI S,DUAN L, et al. Pricing-Driven Service Caching and Task Offloading in Mobile Edge Computing[J]. IEEE Transactions on Wireless Communications,2021,20(7):4495-4512.
- [25] XU H,HUANG W,ZHOU Y, et al. Edge Computing Resource Allocation for Unmanned Aerial Vehicle Assisted Mobile Network With Blockchain Applications[J]. IEEE Transactions on Wireless Communications,2021,20(5):3107-3121.
- [26] WANG Y,SHENG M,WANG X, et al. Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling [J]. IEEE Transactions on Communications,2016,64(10):4268-4282.
- [27] RAZA S,WANG S,AHMED M, et al. Task Offloading and Resource Allocation for IoV using 5G NR-V2X Communication [J]. IEEE Internet of Things Journal,2021,9(13):10397-10410.
- [28] SHANNON C E. A mathematical theory of communication[J]. The Bell System Technical Journal,1948,27(3):379-423.
- [29] SHAH-MANSOURI H,WONG V W S,SCHOBER R. Joint Optimal Pricing and Task Scheduling in Mobile Cloud Computing Systems[J]. IEEE Transactions on Wireless Communications,2017,16(8):5218-5232.
- [30] LIU L,CHANG Z,GUO X, et al. Multiobjective Optimization for Computation Offloading in Fog Computing[J]. IEEE Internet of Things Journal,2018,5(1):283-294.
- [31] ABBAS N,SHARAFEDDINE S,MOURAD A, et al. Joint computing,communication and cost-aware task offloading in D2D-enabled Het-MEC[J]. Computer Networks,2022,209:108900.
- [32] FANG F,XU Y,DING Z, et al. Optimal Resource Allocation for Delay Minimization in NOMA-MEC Networks[J]. IEEE Transactions on Communications,2020,68(12):7867-7881.
- [33] LIU Z,FU J. Resource pricing and offloading decisions in mobile edge computing based on the Stackelberg game[J]. The Journal of Supercomputing,2022,78(6):7805-7824.
- [34] ZHANG H,XIAO Y,CAI L X, et al. A Multi-Leader Multi-Follower Stackelberg Game for Resource Management in LTE Unlicensed[J]. IEEE Transactions on Wireless Communications,2017,16(1):348-361.
- [35] BOYD S,BOYD S P,VANDENBERGHE L. Convex Optimization[M]. Cambridge:Cambridge University Press,2004.
- [36] XIAO Y,BI G,NIYATO D. A Simple Distributed Power Control Algorithm for Cognitive Radio Networks[J]. IEEE Transactions on Wireless Communications,2011,10(11):3594-3600.
- [37] CHEN X,JIAO L,LI W, et al. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing[J]. IEEE/ACM Transactions on Networking,2016,24(5):2795-2808.
- [38] TANG X,REN P,HAN Z. Hierarchical Competition as Equilibrium Program With Equilibrium Constraints Towards Security-Enhanced Wireless Networks [J]. IEEE Journal on Selected Areas in Communications,2018,36(7):1564-1578.
- [39] EBRAHIMZADEH A,MAIER M. Cooperative Computation Offloading in FiWi Enhanced 4G HetNets Using Self-Organizing MEC [J]. IEEE Transactions on Wireless Communications,2020,19(7):4480-4493.
- [40] ADHIKARI M,MUKHERJEE M,SRIRAMA S N. DPTO: A Deadline and Priority-Aware Task Offloading in Fog Computing Framework Leveraging Multilevel Feedback Queuing[J]. IEEE Internet of Things Journal,2020,7(7):5773-5782.



CHEN Xuzhan, born in 1998, postgraduate, is a member of China Computer Federation. His main research interests include edge computing and game theory.



LIN Bing, born in 1986, Ph.D, associate professor, postgraduate supervisor, is a member of China Computer Federation. His main research interests include cloud computing and intelligent computing and its application.