

基于双层 Stackelberg 博弈的 MEC 计算卸载方案*

孙伟峰¹, 张渊魁², 江贺¹, 秦一星¹

¹(大连理工大学 软件学院, 辽宁 大连 116620)

²(中国银行 软件中心, 北京 100094)

通信作者: 孙伟峰, E-mail: wfsun@dlut.edu.cn



摘要: 多接入边缘计算 (multi-access edge computing, MEC) 中的计算卸载问题已经成为当前研究的热点之一。目前的计算卸载方案仅考虑云、边、端结构中的计算卸载问题, 而未考虑到其公、私有云的属性。提出了一种新的计算卸载方案, 所提方案考虑了边缘计算中公有云与私有云之间的关系, 将公有云作为私有云资源的补充, 可以缓解由于私有云资源局限性带来的算力不足问题; 并通过建立双层 Stackelberg 博弈来解决计算卸载问题。对公有云、私有云以及用户的策略和收益进行了分析, 求出了各参与人的最优策略, 证明了双层博弈的纳什均衡解的存在性及唯一性。仿真结果和分析也验证了基于双层 Stackelberg 博弈的计算卸载方案的可行性, 且相较于基于单层 Stackelberg 博弈的卸载方案更高效, 更适合可扩展的边缘计算的环境。

关键词: 多接入边缘计算; 计算卸载; 双层 Stackelberg 博弈

中图法分类号: TP393

中文引用格式: 孙伟峰, 张渊魁, 江贺, 秦一星. 基于双层 Stackelberg 博弈的 MEC 计算卸载方案. 软件学报, 2023, 34(9): 4275-4293. <http://www.jos.org.cn/1000-9825/6642.htm>

英文引用格式: Sun WF, Zhang YK, Jiang H, Qin YX. Computation Offloading Scheme Based on Two-layer Stackelberg Game for Multi-access Edge Computing. Ruan Jian Xue Bao/Journal of Software, 2023, 34(9): 4275-4293 (in Chinese). <http://www.jos.org.cn/1000-9825/6642.htm>

Computation Offloading Scheme Based on Two-layer Stackelberg Game for Multi-access Edge Computing

SUN Wei-Feng¹, ZHANG Yuan-Kui², JIANG He¹, QIN Yi-Xing¹

¹(School of Software Technology, Dalian University of Technology, Dalian 116620, China)

²(Software Center, Bank of China, Beijing 100094, China)

Abstract: The computation offloading problem of multi-access edge computing (MEC) has become one of the research focuses. The current computation offloading scheme only considers the computation offloading problem in the cloud, edge, and end structures and does not take into account the attributes of the public and private clouds. In this study, a novel computation offloading scheme is proposed, which considers the relationship between the public cloud and private cloud in edge computing and regards the public cloud as a supplement to private cloud resources to alleviate the insufficient computing power caused by the limitations of private cloud resources. Moreover, a two-layer Stackelberg game is established to solve the computation offloading problem. The optimal strategies of each player are obtained upon the analysis of the strategies and profits of the public cloud, the private cloud, and users, and the existence and uniqueness of the Nash equilibrium solution to the two-layer game are proved. The simulation results and analysis verifies the feasibility of the computation offloading scheme based on the two-layer Stackelberg game. Compared with the computation offloading scheme based on the single-layer Stackelberg game, the proposed scheme is more efficient and more suitable for edge computing environments.

Key words: multi-access edge computing (MEC); computation offloading; two-layer Stackelberg game

* 基金项目: 国家重点研发计划 (2018YFB1700100); CERNET 创新工程 (NGII20190801); 中央高校基本科研专项 (DUT21LAB115)
收稿时间: 2021-05-25; 修改时间: 2021-08-25, 2021-12-27; 采用时间: 2022-01-12; jos 在线出版时间: 2022-06-06
CNKI 网络首发时间: 2022-12-19

1 引言

MEC 在更加靠近用户的地方进行处理和存储工作,可以显著降低网络转发的数据量,使得效率提高^[1],并且减少了核心网络的负载^[2].其主体结构包括中心云、边缘服务器和终端设备 3 大部分.MEC 服务器是边缘计算中的私有云(包括边缘云和中心云),由于 MEC 服务器的资源是有限的,所以更加需要合理的调配资源.计算卸载模型主要可以分为二态计算卸载和部分计算卸载模型,二态计算卸载假设任务不可再分,只能选择在本地、边缘服务器或中心云执行;而部分计算卸载模型则认为任务可以划分为很多的小任务,选择将一部分任务卸载至服务器.其中 Ren 等人^[3]开发了一个新的部分计算卸载模型,将优化分配通信和计算资源来构造加权和延迟最小化问题转化为一个分段凸优化问题,并提出了一种次梯度算法来寻找最优资源分配解,得出部分计算卸载模型可以获得更好的性能.Mahmood 等人^[4]采用 WPMEC 对部分卸载方案进行了介绍和分析,从整体网络计算能量效率的角度评估了所提出的解决方案的性能.Tuong 等人^[5]提出了一种名为 ACDQN 的深度强化学习算法,该算法考虑了部分计算卸载,用户可以分割计算任务,使一些任务在本地终端上执行,而另一些任务卸载到 MEC 边缘服务器.

MEC 边缘服务器的计算资源和能力是有限的,所以需要一种方法可以判断任务是否卸载至远程服务器,或者卸载至哪一个边缘服务器或者中心云,目前计算卸载方案研究存在的问题主要包括:

(1) 当短时期的任务量暴增时,如双十一或黑色星期五等特殊情况时,单纯的私有云无法满足此时的大量任务,而如果仅扩大私有云的规模会导致在平时的资源浪费,这时就需要考虑到租借公有云的问题.而目前的研究多集中在 MEC 本身结构下的计算卸载问题,而很少有研究考虑到公有云和私有云的问题.

(2) 公有云资源提供商想要更多的利润,MEC 中的中心云和各边缘服务器作为私有云也想要更多的利润,而用户想要更好的服务质量.公有云与私有云之间存在交易关系,而中心云和各边缘服务器之间存在竞争关系,私有云与用户之间也存在交易关系,如何建立一个合理的数学模型,求解出一个各方都满意的计算卸载方案也是当前存在的问题.

本文针对以上问题,提出了一种新的基于双层 Stackelberg 博弈的 MEC 计算卸载方案,该方案采用了部分计算卸载方式,考虑了公、私有云.MEC 中的中心云和各边缘服务器在双层 Stackelberg 博弈中作为私有云,处于公有云与用户的中间位置,用户无法直接和公有云进行交互,需要通过私有云才能将任务卸载至公有云;而公有云作为了私有云资源的补充,可以分布式地解决 MEC 的计算卸载问题.在此结构上通过双层 Stackelberg 博弈描述了公有云、私有云、用户之间的 3 层博弈过程,且对建立的模型进行求解,并对提出的计算卸载方案进行了仿真实验验证本文提出的基于双层 Stackelberg 博弈的计算卸载方案的合理性、可行性和高效性.

本文的主要贡献如下.

(1) 提出了一种新的基于双层 Stackelberg 博弈的 MEC 计算卸载方案,采用了部分卸载的方式,且考虑了公、私有云,解决了 MEC 中私有云资源不足的问题.

(2) 求解了该双层 Stackelberg 博弈中各参与人的最优策略,证明了纳什均衡的存在性与唯一性.

(3) 通过对不同参数下各参与人的收益函数的对比,验证了该方案的合理性与可行性,并通过对比基于单层 Stackelberg 博弈的实验证明了该方案的高效性.

本文第 2 节对本文的相关工作进行介绍,总结了针对 MEC 计算卸载问题的不同方案及各种方案的特点;第 3 节提出了一种新的基于双层 Stackelberg 博弈的 MEC 计算卸载方案,并对方案进行了求解;第 4 节针对第 3 节提出的方案进行实验及评估,证明了该方案的可行性与高效性;最后是总结及展望.

2 相关工作

目前,研究人员提出了很多的计算卸载方法,其主要集中在群体智能、机器学习、强化学习、博弈论等方面上,有部分研究^[6-8]采用了遗传算法(GA)解决计算卸载问题,其中,Zhao 等人^[6]研究了一种基于李亚普诺夫方法的动态优化卸载算法(LODCO)对每个移动设备使用一个扰动参数,通过遗传算法和贪婪算法提出了多用户多服务器的方法,以达到系统的最优结果.Huang 等人^[7]则是着重考虑了安全与能耗的多目标问题,解决异构移动边缘

服务器中的计算卸载. Goudarzi 等人^[8]对遗传算子进行了改进, 减少了无效解. 还有些研究采用了粒子群方法如文献 [9–12]. 其中, Liu 等人^[9]使用了有向无环图 (DAG) 描述任务, 为了在延迟和可靠性之间做出权衡, 他们还提出了一种整数粒子群算法 (ISPO). Huynh 等人^[10]针对了计算卸载与资源分配的联合优化问题, 采用了一种低复杂度、高效率的粒子群算法来解决他们提出的非线性问题并保证其收敛性. Wang 等人^[11]主要考虑卸载任务的响应时间. 粒子群算法是一种元启发式优化技术, 适用于解决环境信息不完全和计算能力有限的连续问题. 因此, 该技术适用于在不保证全局最优解的情况下卸载具有动态行为的生态系统优化问题, 但是 PSO 是一种随机优化技术, 求解依赖于生成的随机变量^[12].

本文用博弈论来解决卸载问题, 而也有一些研究采用深度神经学习、深度学习或强化学习来解决卸载问题. 首先, 一些研究^[13–15]采用了深度神经网络 (DNN) 如. 其中, Shin 等人^[13]研究了资源受限环境下设备时按需卸载问题, 使用 DNN 执行图上客户端和服务器之间的最短路径来划分 DNN 层据此提出了一个新的基于效率的分区算法. 降低了延迟和卸载开销. Jeong 等人^[14]提出了一种增量卸载的神经网络计算 (IONN) 模型, 是边缘计算环境中一种新的基于图的算法. Huang 等人^[15]则是将带宽的卸载和分配策略制定为一个复杂的混合整数规划问题, 使用了 DNN 求解. 有些研究^[16,17]采用了深度学习 (DNQ) 如, Huang 等人^[16]主要是基于 DQN 来解决计算卸载与资源分配的联合优化问题. 同样适用 DQN 的还有 Chen 等人^[17]. 还有些研究^[18–22]采用了深度强化学习 (DRL), 如: Yang 等人^[18]采用 DRL 把混合整数非线性规划变成了一个 RL 问题, 找到最优解. Lu 等人^[19]还利用了 DRL 算法和 LSTM 网络层, 解决了 DRL 的缺陷. 同样使用 DRL 的还有^[20–22]. 而这些方法大多是集中式的, 需要有一个统一的调度器, 而当该调度器的周围网络环境波动时或调度器本身受到损害时, 对于系统有很大的影响.

博弈论^[23]因其分布式、可保证均衡等特性也成为目前研究的热门方法经典博弈论一般可以分为合作博弈和非合作博弈, 前者更适合于更重视公平性的合作环境. 合作博弈通过用户之间预先约定的协议来处理用户之间的协作这种合作会影响用户的选择以及影响用户的效用. 这类博弈的特点是广泛界定和隐含的规则、强调的联盟和承诺的可用性. 一些研究^[24,25]采用了合作博弈如, Mukherjee 等人^[24]和 Guo 等人^[25]采用了合作博弈来解决计算卸载问题. 还有一些研究^[26–30]采用非合作博弈试图达到 Stackelberg 或 Nash 均衡作为解, 如 Chen^[26]将移动设备用户间的分散计算卸载决策问题表述为一个分散计算卸载博弈, 提出了一种满足移动云计算环境中有效的分散计算卸载的方法. 为了改进移动设备能耗和时延的联合优化, Jošilo 等人^[27]提出了用 Stackelberg 博弈解决无线和计算资源分配问题, 考虑运营商之间的交互管理器 (追随者) 的设备和装置 (领导人), 作为决策者是否将使用共享资源. Liang 等人^[28]通过研究定价策略来解决 MEC 的计算卸载问题, 并首次探讨了非线性定价策略, 期间采用了 Stackelberg 博弈, 最终通过分析和松弛等方法导出了低阶问题的最优解, 求解了该非凸的问题. Lei 等人^[29]引入了演化博弈框架, 提出了一种新的基于概率性能感知和演化博弈理论的混合云边缘环境计算卸载方法. Zhou 等人^[30]则是利用了非合作博弈解决部分卸载问题, 在此基础上, 他们给出了一种低时间复杂度的局部计算卸载方法实现纳什均衡.

3 基于双层 Stackelberg 博弈的计算卸载方案

3.1 问题描述

计算卸载问题主要需要解决的是: 终端设备的任务是否卸载至远程服务器, 或者卸载至哪一个边缘服务器或者中心云. 本文假设任务对于不同资源类型有着不同的需求, 任务可以按照所需求的资源类型进行拆分, 而任务的相同资源类型需求的部分为原子的, 不可再拆分的. 如图 1 带公有云的 MEC 计算卸载架构所示, 将 MEC 中的边缘服务器和中心云服务器统称为私有云.

不同于传统的计算卸载模型, 本文考虑到: MEC 中任务量的变化较大, 例如双十一、黑五或游戏福利等活动时, 任务量会短时间内变大, 而单纯的私有云可能无法满足需求; 此时如果仅考虑增加私有云服务又会导致成本较高. 故本文考虑到将公有云加入计算卸载系统, 私有云可以通过购买公有云的服务来应对任务量激增的情况, 而大部分 MEC 计算卸载方案没有考虑到共、私有云的问题. 如图 1 带公有云的 MEC 计算卸载架构所示, 其中, 中心

云服务器与边缘云服务器的主要差别在于资源量与延迟, 于是将 MEC 中的边缘云服务器和中心云服务器统称为私有云. 设计用户无法直接和公有云进行交互, 需要通过私有云才能将任务卸载至公有云.

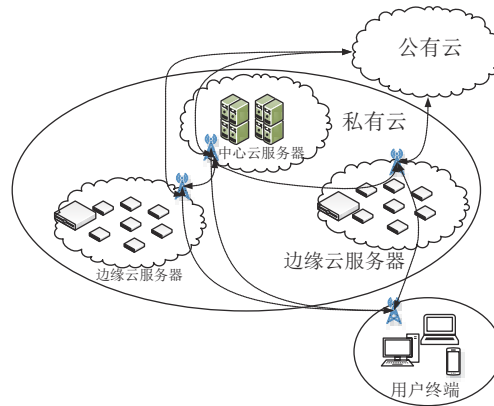


图1 带公有云的 MEC 计算卸载架构

3.2 模型建立

3.2.1 模型结构及数学符号

公有云和各边缘服务器之间存在买卖关系, 公有云为卖方, 而各私有云为买方. 公有云想要将资源以更高的价格进行卖出, 而私有云想要以更低的价格买入. 而私有云与用户之间也存在博弈, 用户们想要更好的服务及更低廉的价格, 而私有云想要更高的收益. 由于上述存在的这两种关系, 本文提出了一种基于双层 Stackelberg 博弈的计算卸载方案. 在该方案中, 公有云、私有云以及用户作为博弈的参与者, 被定义为理性的独立个体. 为了便于阅读以及理解, 表 1 中归纳并总结了本节主要的数学符号和具体的含义.

表 1 数学符号

| 符号 | 含义 |
|------------|--|
| S | 任务流集合 $S = \{1, 2, \dots, S\}$ |
| N | 服务器集合 $N = \{1, 2, \dots, N\}$, 其中 $j = N + 1$ 时表示在本地执行 |
| Ω | 资源类型集合 $\Omega = \{1, 2, \dots, m\}$ |
| λ | 卸载概率参数 |
| y_r | 任务流对于资源类型 r 需求的任务量 |
| R_r^j | 服务器 j 上, 资源类型为 r 的现存资源总量 |
| R_r^C | 公有云上, 资源类型为 r 的现存资源总量 |
| t_r^C | 任务流的资源类型 r 的部分在公有云上用时 |
| t_r^j | 任务流的资源类型 r 的部分在边缘服务器 j 上用时 |
| P_r^C | 公有云上资源类型 r 的功率 |
| P_j^{TC} | 服务器 j 至公有云的传输功率 |
| P_r^j | 服务器 j 上资源类型 r 的功率 |
| E_r^j | 任务流在服务器 j 上的总能耗 |
| E_r^{Cj} | 任务流的资源类型 r 的部分从服务器 j 运输到公有云上的能耗 |
| q_r^j | 任务流的资源类型 r 的部分在服务器 j 上的服务质量 |
| q_r^C | 任务流的资源类型 r 的部分在公有云上的服务质量 |

表 1 数学符号 (续)

| 符号 | 含义 |
|----------------------|----------------------------------|
| ξ | 单位能耗的价格 |
| t_r^{bound} | 任务流的资源类型 r 的部分能接受的用时长限 |
| p_r^{bound} | 任务流的资源类型 r 的部分能接受的报价上限 |
| f_r^j | 服务器 j 对公有云的资源类型 r 的报价策略 |
| g_r^j | 服务器 j 对公有云的资源类型 r 的资源量需求策略 |
| b_r^j | 公有云对服务器 j 的分配策略 |
| p_r^j | 服务器 j 对于任务流 i 关于资源类型 r 的报价 |
| a_r^j | 任务流对边缘服务器 j 的资源类型 r 的分配策略 |
| ϕ_r | 公有云的资源类型 r 出售给其他项目的平均收益 |

第 1 层 Stackelberg 博弈的流程如图 2 左侧两个实体的交互, 公有云首先将其资源类型、运行效率以及传输效率等基础信息发送至各私有云中, 然后各边私有云作为 Stackelberg 博弈中的领导者根据公有云所给的信息以及自身需求向公有云提供自身对各类资源的需求量以及对应的单位资源量报价策略, 其中各私有云服务器之间存在竞争, 且需考虑到公有云如果不将资源卖给私有云依然可以通过资源创造其他收益, 此时公有云会按照自身利益最大化原则为各私有云服务器提供资源策略。

第 2 层 Stackelberg 博弈的流程如图 2 右侧两个实体的交互, 假设某个用户先将任务流的信息发送给各个私有云服务器, 这时各个私有云服务器通过拟资源分配, 确定如果接受该任务后可以提供给该任务的任务完成时间 t_r^j ($i \in S, j \in N$), 能耗 E_r^j ($i \in S, j \in N$), 资源类型及总量 R_r^j ($j \in N, r \in \Omega$) 等信息. 作为 Stackelberg 博弈的领导者, 边缘服务器根据这些信息推测可以给与任务流 i 的服务质量以及报价 p_r^j ($j \in N, r \in \Omega$) 并发送给该用户. 用户作为 Stackelberg 博弈的跟随者再根据自身利益最大化原则选择合适的计算卸载策略 a_r^j ($j \in N, r \in \Omega$).

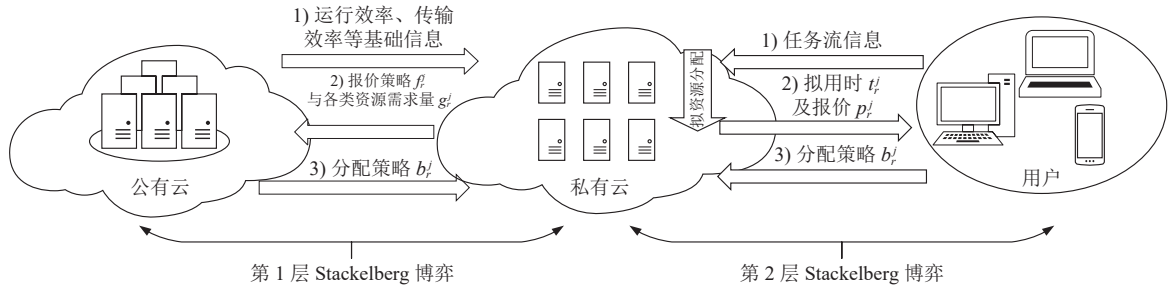


图 2 双层 Stackelberg 博弈

3.2.2 服务质量与共同知识

在用户选择服务器时, 一个重要的指标就是服务质量. 用户可以通过服务质量来衡量是否要卸载与卸载到哪个服务器. Mahmood 等人^[4]根据用户设备的服务质量指标要求对设备进行优先级排序; Gao 等人^[31]通过将计算密集型的工作负载转移到边缘服务器, 可以显著降低移动设备的能量消耗, 从而大大提高服务质量指标.

由于先前工作中, 大部分并没有使用服务质量指标, 而少数使用的服务质量指标并没有考虑到本文提出的公有云、私有云、用户三者之间的卸载博弈, 不适用于本文, 故本文提出了一个服务质量指标, 如公式 (1), 该服务质量指标针对任意任务流 i ($i \in S$). 其中考虑到了边缘云或中心云 j 给用户的报价以及资源完成这个任务流的拟用时. t_r^j 主要通过用户给的任务基本信息, 服务器自身进行拟资源分配, 得到处理时间再结合网络延迟组成.

$$q_r^j(p_r^j, t_r^j) = \begin{cases} 0, & \text{如果 } (t_r^j > t_r^{\text{bound}} \text{ 或 } p_r^j > p_r^{\text{bound}}) \\ \frac{(p_r^{\text{bound}} - p_r^j)(t_r^{\text{bound}} - t_r^j)}{p_r^{\text{bound}} \cdot t_r^{\text{bound}}}, & \text{其他} \end{cases} \quad (1)$$

其中, t_r^{bound} 和 p_r^{bound} 分别表示任务流 i 对于资源类型 r 能接受的用时上限和能接受的报价上限, $t_r^{\text{bound}} > 0, p_r^{\text{bound}} > 0$. 可以看出当用时或报价不满足任务流的需求时, 服务质量为 0, 而当服务器报价更低时或者拟用时更短时, 服务质量则会更高.

对于公式 (1), 这些参数都是真实可得到的, 所以通过图像来进一步分析服务质量指标关于报价以及拟用时的变化趋势, 进而证明了服务质量指标的可行性. 如图 3 与图 4 所示, 一方面, 随着报价以及拟用时的上升, 服务质量指标会随之快速的下降, 即当用户被索取更多的报价时, 会通过缩短用时 t_r^j 来提高服务质量; 另一方面, 服务质量指标会随着报价和拟用时的下降而上升, 即当用户有更多的任务或选取边缘服务器为其提供服务时, 用户需要提高报价以确保任务的完成. 在同等条件下, 当服务器报价更低以及拟用时更短时, 其服务质量会更高.

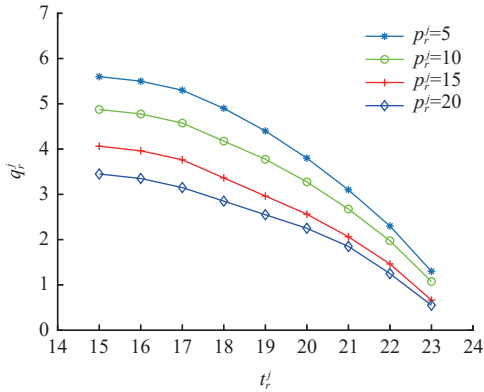


图 3 p_r^j 对服务质量指标的影响

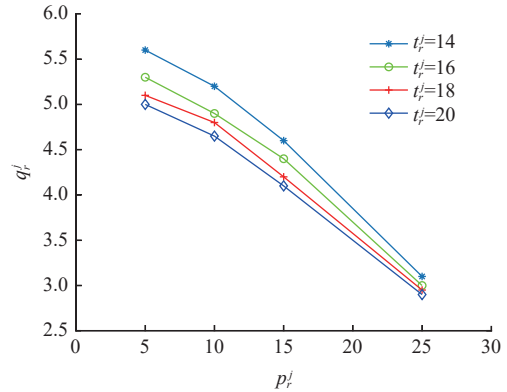


图 4 t_r^j 对服务质量指标的影响

共同知识是每个参与人都知道的, 且每个人都知道其他人知道这个共同知识, 且每个人都知道每个人都知道其他人知道这个共同知识, ... 在本文的双层 Stackelberg 博弈的各个参与人中, 存在共同知识, 卸载概率 $x_r(q_r^j)$, 即在用时为 t_r^j , 且报价为 p_r^j 时, 任务选择卸载到服务器 j 的概率. 假设卸载概率具体表示如公式 (2):

$$x_r(q_r^j) = 1 - e^{-\lambda q_r^j} \tag{2}$$

其中, $\lambda > 0$, 服务器根据 $x_r(q_r^j)$, 再根据服务器自身情况做出决策. 通过图 5 可以看出报价 p_r^j 和 t_r^j 对于卸载概率的影响, 图中令 $\lambda = 5$, 表示了服务质量从 0.1 到 0.8 时卸载概率. 可以看出, 随着服务质量从小变大, 卸载概率在不断增长, 且增长逐渐缓慢, 趋近于 1.

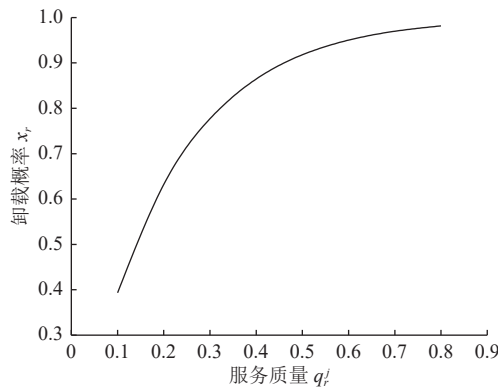


图 5 卸载概率与服务质量之间的关系

3.2.3 公有云策略模型

本文提出的计算卸载方案中公有云起到关键作用, 它可以对于 MEC 的资源进行一定的补充, 来应对不同的环

境, 尤其是当任务量较大的时候. 而公有云显然不会无限制地对私有云或边缘服务器出售自己的资源, 公有云如果不将资源出售给私有云或边缘服务器, 而是投于各类其他项目服务, 这时也会有收益, 这就是资源本身的潜在收益. 显然只有当出售给私有云资源更有利时才会出售. 于是假设在该模型中, 任务流集合为 $S = \{1, 2, \dots, S\}$, 私有云及边缘服务器集合为 b_r^j , 这其中的资源分为不同的类型, 不同的任务对于不同的资源有不同的需求, 设 $\Omega = \{1, 2, \dots, m\}$ 为资源类型集合, 假设公有云当前各资源总量为 $R^C = \{R_1^C, R_2^C, \dots, R_m^C\}$, 其中 R_r^C 为公有云上资源类型为 r 的现存资源总量. 对任意任务流 i 来说, 设公有云的策略集为 $b_r^j (j \in N, r \in \Omega)$ 为公有云为边缘服务器 j 提供资源类型 r 的数量, $b_r^j \leq g_r^j$.

公有云的利润来源主要有两部分, 一部分为将资源出售给其他项目的利润, 一部分为将资源出售给私有云的利润. 假设资源类型 r 出售给其他项目的平均收益为 φ_r . 则定义公有云总利润为 $u^C(b_r^j)$, 如公式 (3):

$$u^C(b_r^j) = U_E^C(b_r^j) + U_O^C(b_r^j) \quad (3)$$

其中, $U_E^C(b_r^j)$ 为将资源出售给私有云的总利润, $U_O^C(b_r^j)$ 为将资源出售给其他项目的总利润. 再考虑到能耗问题以及能源花销, 设 E_r^{Cj} 为公有云租用给私有云服务器 j 的部分资源类型为 r 的总能耗. 显然 E_r^{Cj} 可以分为两个部分, 一部分是执行能耗, 另一部分是传输能耗.

$$E_r^{Cj} = P_r^C \cdot t_r^C + \frac{P^{jC} \cdot d}{v^{jC}} \quad (4)$$

$$v^{jC} = B \log_2 \left(1 + \frac{P^{jC} \cdot g^2}{N_0 B} \right) \quad (5)$$

其中, $P_r^C (r \in \Omega)$ 为公有云资源类型 r 的功率, t_r^C 为公有云资源类型 r 的运行时间, $P^{jC} (j \in N)$ 为由边缘服务器 j 至公有云的传输功率, d 为数据大小, v^{jC} 为由边缘服务器 j 至公有云的传输速率. B 为传输平均无线信道带宽, g 为信道增益, N_0 表示高斯白噪声的单边功率谱密度. 于是得到将资源出售给边缘服务器的总利润 U_E^C , 具体如公式 (6):

$$U_E^C(b_r^j) = \sum_{j \in N} \sum_{r \in \Omega} b_r^j (f_r^j - \zeta \cdot E_r^{Cj}) \quad (6)$$

其中, f_r^j 为服务器 j 对资源类型 r 的单位报价策略, $\zeta \cdot E_r^{Cj}$ 为公有云为边缘服务器 j 提供资源类型 r 能耗的花销, ζ 为单位能耗的价格. 此外, 将资源出售给其他项目的总利润 $U_O^C(b_r^j)$ 如公式 (7):

$$U_O^C(b_r^j) = \sum_{r \in \Omega} \varphi_r (R_r^C - \sum_{j \in N} b_r^j) + \sum_{j \in N} \sum_{r \in \Omega} (f_r^j - \varphi_r) \cdot \ln(1 + g_r^j - b_r^j) \quad (7)$$

其中, $\varphi_r (r \in \Omega)$ 为公有云在其他项目的资源类型为 r 的单位收益. $\sum_{j \in N} \sum_{r \in \Omega} (f_r^j - \varphi_r) \cdot \ln(1 + g_r^j - b_r^j)$ 为规模效应带来的额外收益. 综上, 得到公有云的优化模型, 如公式 (8a)–公式 (8c):

$$\max_{b_r^j} U^C(b_r^j) \quad (8a)$$

$$\text{s.t. } b_r^j \leq g_r^j, \forall r \in \Omega, \forall j \in N \quad (8b)$$

$$b_r^j \geq 0, \forall r \in \Omega, \forall j \in N \quad (8c)$$

其中, 目标函数公式 (8a) 表示公有云希望最大化自己的利润, 根据利润来选择策略 b_r^j , 约束条件 (8b) 表示公有云针对资源类型 r , 给予各个私有云的资源量 b_r^j 不得多于各私有云索取的资源量; 约束条件 (8c) 表示公有云针对资源类型 r , 给予各个私有云的资源量 b_r^j 必须大于零.

3.2.4 私有云策略模型

私有云主要由各个边缘云服务器和中心云服务器构成, 主要依靠向用户出售服务来获取利润, 并且当某个服务器 (边缘服务器或中心服务器) 的资源不足时, 还需要向公有云购买资源. 每个服务器都是理性的, 都是以自身利益最大化为目标, 赚取更多的利润, 而且各服务器之间是只有竞争关系而没有合作关系. 设边缘服务器与中心云服务器的集合为 $N (1, 2, \dots, N)$, 则任意服务器 $j \in N$, 其当前各资源总量为 $R^j = (R_1^j, R_2^j, \dots, R_m^j)$, 并且本次交易购买资源的总预算为 M^j . 服务器 j 的策略主要为向公有云购买资源时的报价策略 $f^j = \{f_1^j, f_2^j, \dots, f_m^j\}$, 向公有云购买时的各类型资源的需求 $g^j = \{g_1^j, g_2^j, \dots, g_m^j\}$ 另一方面为向用户出售资源时的报价策略 $p^j = \{p_1^j, p_2^j, \dots, p_m^j\}$. 于是得出,

在第 1 层 Stackelberg 博弈中, 私有云 j 的收益如公式 (9):

$$U_1^j(f^j, g^j, p^j) = \sum_{r \in \Omega} [x_r(q_r^c) \cdot b_r^j \cdot p_r^j - f_r^j \cdot b_r^j] \quad (9)$$

其中, $\sum_{r \in \Omega} x_r(q_r^c) \cdot b_r^j \cdot p_r^j$ 为购买资源的价值, b_r^j 为公有云的分配策略, 与 g^j 相关. $\sum_{r \in \Omega} f_r^j \cdot b_r^j$ 为购买资源的花费.

在第 2 层 Stackelberg 博弈中, 私有云 j 的收益如公式 (10):

$$U_2^j(f^j, p^j) = U_C^j(f^j, p^j) + U_u^j(f^j, p^j) \quad (10)$$

其中, $U_C^j(f^j, p^j)$ 为从公有云获取的资源的收益, $U_u^j(f^j, p^j)$ 为从用户获取的收益. 展开如公式 (11) 和公式 (12):

$$U_C^j(f^j, p^j) = \sum_{r \in \Omega} x_r(q_r^c) \cdot y_r \cdot p_r^j - f_r^j \cdot y_r \quad (11)$$

$$U_u^j(f^j, p^j) = \sum_{r \in \Omega} a_r^j \cdot (p_r^j - C_E^j(y_r)) \quad (12)$$

其中, a_r^j 为用户购买资源策略, b_r^j 为公有云卖给服务器的策略, y_r 为任务对于资源类型 r 需求的总大小, $x_r(q_r^c)$ 为公有云提供的服务质量的卸载概率. 设 E_r^j 为边缘服务器 j 的资源类型为 r 的总能耗. 显然 E_r^j 可以分为两个部分, 一部分是执行能耗, 另一部分是传输能耗, 如公式 (13):

$$E_r^j = p_r^j \cdot t_r^j + \frac{p^j \cdot d}{v^j} \quad (13)$$

$$v^j = B \log_2 \left(1 + \frac{p^j \cdot g^2}{N_0 B} \right) \quad (14)$$

其中, $t_r^j (r \in \Omega)$ 为服务器 j 的资源类型 r 的功率, t_r^j 为公有云资源类型 j 的运行时间, $p^j (j \in N)$ 为任务流传送至服务器 j 的传输功率, d 为数据大小, v^j 为任务流传送至服务器 j 的传输速率. B 为传输平均无线信道带宽, g 为信道增益, N_0 表示高斯白噪声的单边功率谱密度. 于是得到 $C_E^j(y_r)$ 如公式 (15):

$$C_E^j(y_r) = y_r \cdot \zeta \cdot E_r^j \quad (15)$$

其中, $\zeta \cdot E_r^j$ 为公有云为边缘服务器 j 提供资源类型 r 能耗的花销, ζ 为单位能耗的价格. 此时得到私有云的第 1 层 Stackelberg 博弈的模型如公式 (16a)–公式 (16e):

$$\max_{f^j, p^j, g^j} U_1^j(f^j, p^j, g^j) \quad (16a)$$

$$\text{s.t. } \sum_{r \in \Omega} f_r^j \leq M^j, \forall r \in \Omega \quad (16b)$$

$$f_r^j > \varphi_r, \forall r \in \Omega \quad (16c)$$

$$p_r^j \geq 0, \forall r \in \Omega \quad (16d)$$

$$g_r^j \geq 0, \forall r \in \Omega \quad (16e)$$

其中, 目标函数 (16a) 表示私有云在第 1 层 Stackelberg 博弈中基于利益最大化原则选择策略, 约束条件 (16b) 表示服务器 j 从公有云服务购买各类资源花费之和不得超过服务器 j 的总预算; 约束条件 (16c) 表示服务器 j 给公有云的报价需要大于公有云的资源类型 r 出售给其他项目的平均收益; 约束条件 (16d) 表示服务器 j 对于任务流 i 关于资源类型 r 的报价需大于等于零; 约束条件 (16e) 服务器 j 对公有云的资源类型 r 的资源量需求策略需要大于等于零. 类似的, 第 2 层 Stackelberg 博弈的模型可以表示为公式 (17a)–公式 (17d):

$$\max_{f^j, p^j} U_2^j(f^j, p^j) \quad (17a)$$

$$\text{s.t. } \sum_{r \in \Omega} f_r^j \leq M^j, \forall r \in \Omega \quad (17b)$$

$$p_r^j \geq 0, \forall r \in \Omega \quad (17c)$$

$$f_r^j > \varphi_r, \forall r \in \Omega \quad (17d)$$

其中, 目标函数 (17a) 表示私有云在第 2 层 Stackelberg 博弈中基于利益最大化原则选择策略, 其余约束条件类似第 1 层 Stackelberg 博弈.

3.2.5 用户策略模型

用户希望自己的任务流可以有更高的服务质量, 这时可以选择将计算卸载至本地或者选择上传到服务器. 以任务流 $i (i \in S)$ 为例, 用户希望每个任务流的服务质量都可以最大化, 设用户对于任务流 i 的策略为 $a_r^i (r \in \Omega, j \in N \cup \{N+1\})$, 表示用户将任务流的资源类型 r 的部分分配给了服务器 j , 这时 $j = N+1$ 表示在本地执行. 此时 $a_r^i = 0$ 或 $a_r^i = y_r$, y_r 为任务流的资源类型 r 的部分总需求, 即需要满足 $y_r = \sum_{j \in N \cup \{N+1\}} a_r^i$.

假设用户的任务流 $i (i \in S)$ 在本地资源类型 r 上执行的能耗为 $E_r^i (r \in \Omega)$, 则任务流在本地执行的总能耗为 E^i , 将其表示为公式 (18):

$$E_r^i = p_r^i \cdot t_r^i \quad (18)$$

其中, $p_r^i (r \in \Omega)$ 为本地资源类型 r 执行一个 CPU 周期的功耗. $t_r^i (r \in \Omega)$ 为任务流 i 在本地资源 r 上执行的时间, 用 CPU 周期数描述, 单位为 cycles.

这时用户对于任务流 i 的收益如公式 (19):

$$U_i(a_r^i) = \sum_{j \in N \cup \{N+1\}} \sum_{r \in \Omega} a_r^i \cdot q_r^i(p_r^j, t_r^j) \quad (19)$$

其中, (p_r^j, t_r^j) 为服务质量. 此外, 当 $j = N+1$ 时, 即在本地执行时, 所需价格为能耗花销, 如公式 (20):

$$p_r^{N+1} = \zeta \cdot E_r^i \quad (20)$$

其中, ζ 为单位能耗的价格. 且得到用户对于任务流 i 的优化模型可以表示为公式 (21a)–公式 (21d):

$$\max_{a_r^i} U_i(a_r^i) \quad (21a)$$

$$\text{s.t. } y_r = \sum_{j \in N \cup \{N+1\}} a_r^i, \forall j \in N \cup \{N+1\}, r \in \Omega \quad (21b)$$

$$q_r^i(p_r^j, t_r^j) > 0, \forall j \in N \cup \{N+1\}, r \in \Omega \quad (21c)$$

$$a_r^i \geq 0, \forall j \in N \cup \{N+1\}, r \in \Omega \quad (21d)$$

约束条件 (21b) 表示任务流对于任意资源类型 r 的分配策略符合该任务流的资源类型 r 的部分总需求. 约束条件 (21c) 表示分配结果需使得服务质量大于零, 即满足该任务的需求.

3.3 最优策略与均衡分析

本节主要求出双层 Stackelberg 博弈中的各个参与人的最优策略, 以及分析其中的纳什均衡的存在性以及求出纳什均衡解.

3.3.1 公有云的最优策略

公有云的利润主要来源于两部分: 出售给其他项目的利润和出售给私有云的利润, 由于其作为第 1 层 Stackelberg 博弈的跟随者, 会先收到来自私有云的报价策略 $f_r^j (j \in N, r \in \Omega)$ 以及其对于各类资源的需求量 $g_r^j (j \in N, r \in \Omega)$, 公有云作为一个理性人根据 f_r^j, g_r^j 来选择自己的最优策略. 根据公式 (7), 得到公有云的效用函数如公式 (22):

$$U^C(b_r^j) = \sum_{j \in N} \sum_{r \in \Omega} b_r^j (f_r^j - \zeta E_r^{Cj}) + \sum_{r \in \Omega} \varphi_r (R_r^C - \sum_{j \in N} b_r^j) + \sum_{j \in N} \sum_{r \in \Omega} (f_r^j - \varphi_r) \cdot \ln(1 + g_r^j - b_r^j) \quad (22)$$

关于 b_r^j 求一阶偏导以及二阶偏导, 得公式 (23) 及公式 (24):

$$\frac{\partial U^C(b_r^j)}{\partial b_r^j} = (f_r^j - \varphi_r) \left(\frac{-1}{1 + g_r^j - b_r^j} + 1 \right) - \zeta E_r^{Cj} \quad (23)$$

$$\frac{\partial^2 U^C(b_r^j)}{\partial b_r^j{}^2} = \frac{(\varphi_r - f_r^j)}{(1 + g_r^j - f_r^j)^2} \quad (24)$$

由于 $f_r^j > \varphi_r$, 故而 $\frac{\partial^2 U^C(b_r^j)}{\partial b_r^j{}^2} < 0$, $U^C(b_r^j)$ 关于 b_r^j 是一个凸函数. 那么通过令 $\frac{\partial U^C(b_r^j)}{\partial b_r^j} = 0$, 即可得到 b_r^j 的最优解 $b_r^j(f_r^j, g_r^j)$, 如公式 (25):

$$b_r^j(f_r^j, g_r^j) = \frac{\varphi_r - f_r^j}{f_r^j - \varphi_r - \zeta E_r^{C_j}} + g_r^j + 1 \quad (25)$$

再考虑到约束条件 $0 \leq b_r^j \leq g_r^j$, 可得出结论公有云的策略合集具有非空性、凸性以及非紧性, 故当支付价格过低时, 公有云不会将任务分配给私有云; 而当私有云的支付单价足够多时, 公有云会将所有资源都分配给私有云. 除此之外, 当支付价格趋于稳定时, 公有云的最优策略 b_r^{j*} 最终解如公式 (26):

$$b_r^{j*}(f_r^j, g_r^j) = \begin{cases} 0, & \text{如果 } b_r^j(f_r^j, g_r^j) < 0 \\ g_r^j, & \text{如果 } b_r^j(f_r^j, g_r^j) > 0 \\ \frac{\phi_r - f_r^j}{f_r^j - \phi_r - \zeta E_r^{C_j}} + g_r^j + 1, & \text{其他} \end{cases} \quad (26)$$

对于任意资源, 当支付价格总和处于极值时, 公有云的最优策略就是分配与不分配两种. 由 b_r^{j*} 关于 f_r^j 的一阶偏导和二阶偏导均大于零可知 b_r^{j*} 是关于 f_r^j 的凹增函数, 所以公有云关于资源的分配策略是唯一的. 再借助比例分配共享机制, 按照支付价格比例将 b_r^{j*} 分配给相应的私有云. 又因为任务的任意性以及不同任务的无关联性可知, 每种任务的最优分配策略存在且唯一.

3.3.2 用户的最优策略

用户主要目的是使得任意任务流的服务质量达到最高, 即最大化 $U_i(a_r^i)$, 而用户是第 2 层 Stackelberg 博弈的跟随者, 根据公式 (19), 并且本文假设用户的任务流对于资源 r 的需求部分只能分配给同一个服务器, 即为原子的, 不可再分. 而由于服务器数量有限, 故而用户的策略集为有限集, 必有最优解 a_r^{i*} . 其次, 为了保证所有任务均被分配, 需要对于 $p_r^{\text{bound}}, t_r^{\text{bound}}$ 进行调整, 于是通过算法 1 对用户的最优策略进行求解.

如算法 1 为用户的最优策略算法, 首先找出任务流中那些所有服务器均没法满足其 QoS 需求的部分, 将其放入集合 zeroSet, 并对其他可以满足 QoS 的部分按根据 $y_r \cdot q_r^j$ 最大原则选择计算卸载位置 (第 1–8 行). 然后再将 zeroSet 中的部分进行分配, 通过调整阈值 $p_r^{\text{bound}}, t_r^{\text{bound}}$, 以及 $y_r \cdot q_r^j$ 最大化原则将其卸载 (第 9–18 行).

算法 1. 用户的最优策略.

输入: 私有云报价 p_r^j 及拟用时 t_r^j , 任务信息 y_r , 用户初始报价及用时阈值 $p_r^{\text{bound}}, t_r^{\text{bound}}$;

输出: 修正后的 $p_r^{\text{bound}}, t_r^{\text{bound}}$, 最优策略 a_r^{i*} .

1. **For** r in Ω
 2. **If** $\sum_{j \in \mathcal{N} \cup \{N+1\}} q_r^j(t_r^j, p_r^j) = 0$
 3. zeroSet = zeroSet.append(r);
 4. **Else**
 5. maxnum $\leftarrow y_r \cdot q_r^j$ 在 $j \in \mathcal{N} \cup \{N+1\}$ 上的最大值对应的上标 j ;
 6. $a_r^{\text{maxnum}} \leftarrow y_r$;
 7. **End if**
 8. **End for**
 9. **For** r in zeroSet
 10. **While** $\sum_{j \in \mathcal{N} \cup \{N+1\}} q_r^j(t_r^j, p_r^j) = 0$
 11. $p_r^{\text{bound}} \leftarrow p_r^{\text{bound}} + \epsilon_p$;
 12. $t_r^{\text{bound}} \leftarrow t_r^{\text{bound}} + \epsilon_t$;
 13. **End while**
 14. maxnum $\leftarrow y_r \cdot q_r^j$ 在 $j \in \mathcal{N} \cup \{N+1\}$ 上的最大值对应的上标 j ;
 15. $a_r^{\text{maxnum}} \leftarrow y_r$;
 16. **End for**
-

用户不会向报价高于报价上限的私有云提交任何资源, 以用户 r 为例, $p_r^i < p_r^{\text{bound}}$ 的情况, 且资源提交量会按照报价的提升而严格的下降. 由于用户的任意性以及不同用户之间的无关联性可知, 每一个用户都存在唯一的最优策略 p_r^{i*} .

3.3.3 私有云的最优策略

私有云作为理性人, 依然以自身收益最大化为目的, 而其为第 1 层和第 2 层 Stackelberg 博弈的领导者, 在与公有云博弈时, 即在第 1 层 Stackelberg 博弈时, 其目的为得到最优的报价与资源请求量, 最大化其收益. 在第 2 层 Stackelberg 博弈时, 其主要目的是得到最优的报价策略, 最大化其收益. 先对 p_r^i 的取值进行分析, 求得最优策略. 将第 2 层 Stackelberg 博弈的目标函数展开如公式 (27):

$$U_2^i(f^j, p^j) = \sum_{r \in \Omega} [x_r(q_r^c) \cdot y_r \cdot p_r^j - f_r^j \cdot y_r + x_r(q_r^j) \cdot y_r \cdot (p_r^j - C_E^j(y_r))] \quad (27)$$

考虑 $p_r^j \geq p_r^{\text{bound}}$, $t_r^j \geq t_r^{\text{bound}}$ 时,

$$U_2^i(f^j, p^j) = \sum_{r \in \Omega} x_r(q_r^c) \cdot y_r \cdot p_r^j - f_r^j \cdot y_r + \left[1 - \exp\left(-\lambda \left[\frac{(p_r^{\text{bound}} - p_r^j)(t_r^{\text{bound}} - t_r^j)}{p_r^{\text{bound}} \cdot t_r^{\text{bound}}} \right] \right) \right] \cdot y_r \cdot (p_r^j - (C_E^j(y_r))) \quad (28)$$

为了方便表述, 令

$$\Phi = \frac{t_r^{\text{bound}} - t_r^j}{p_r^{\text{bound}} \cdot t_r^{\text{bound}}} \quad (29)$$

$$\Psi = \frac{t_r^{\text{bound}} - t_r^j}{t_r^{\text{bound}}} \quad (30)$$

此时对 $U_2^i(f^j, p^j)$ 关于 p_r^j 求导, 结果如公式 (31):

$$\frac{\partial U_2^i(f^j, p^j)}{\partial p_r^j} = y_r \cdot [x_r(q_r^c) + 2 + (1 - \exp\{-\lambda \cdot \Phi \cdot p_r^j - \lambda \Psi\}) \cdot (-\lambda \cdot \Phi \cdot p_r^j + 1)] \quad (31)$$

再求关于 p_r^j 二阶导, 结果如公式 (32):

$$\frac{\partial U_2^i(f^j, p^j)}{\partial p_r^j} = y_r \cdot \lambda \cdot \Phi \cdot \exp\{-\lambda \cdot \Phi \cdot p_r^j - \lambda \Psi\} \cdot (2 - \lambda \cdot \Phi \cdot p_r^j) \quad (32)$$

可以看出当 $p_r^j \geq \frac{2}{\lambda \Phi}$ 时 $U_2^i(f^j, p^j)$ 为凸函数, 当 $p_r^j \leq \frac{2}{\lambda \Phi}$ 时 $U_2^i(f^j, p^j)$ 为凹函数. 故取 $0 < \varepsilon < \frac{1}{\lambda \Phi}$ 为无穷小的数时, 取 $p_r^j = \frac{2}{\lambda \Phi} + \varepsilon$, 带入 $\frac{\partial U_2^i(f^j, p^j)}{\partial p_r^j}$ 中, 结果如公式 (33):

$$\frac{\partial U_2^i(p_r^j)}{\partial p_r^j} = y_r [x_r(q_r^c) + 2 + (1 - \exp\{-2 - \lambda \Phi \varepsilon - \lambda \Psi\}) \cdot (-2 - \lambda \cdot \Phi \cdot \varepsilon + 1)] \quad (33)$$

即 $\frac{\partial U_2^i(p_r^j)}{\partial p_r^j} > 0$, 同理, 取 $p_r^j = \frac{2}{\lambda \Phi} - \varepsilon$ 时, 也有 $\frac{\partial U_2^i(p_r^j)}{\partial p_r^j} > 0$. 即函数在 $p_r^j = \frac{2}{\lambda \Phi}$ 周围均为增函数. 那么最优解在极值点 $\frac{\partial U_2^i(p_r^j)}{\partial p_r^j} = 0$ 处或在边界 $p_r^j = 0$ 处, 而显然在 $p_r^j = 0$ 时 $U_2^i(f^j, p^j) < 0$, 并且方程 $\frac{\partial U_2^i(p_r^j)}{\partial p_r^j} > 0$ 与 f^j 的取值无关, 从而可以借助梯度下降法的思路, 求出最优策略 p_r^{i*} , 如算法 2 私有云的最优策略算法.

算法 2. 私有云的最优策略.

输入: $x_r(q_r^c), \Phi, \Psi$;

输出: 最优策略 p_r^{i*} .

1. **While** $\frac{\partial U_2^i(p_r^j)}{\partial p_r^j} > \varepsilon$ && $0 < p_r^j < p_r^{\text{bound}}$
 2. $p_r^j \leftarrow p_r^j + \alpha \frac{\partial U_2^i(p_r^j)}{\partial p_r^j}$;
 3. **End while**
 4. $p_r^{i*} \leftarrow p_r^j$;
-

在求得了最优策略 p_r^j 之后, 考虑求出第 1 层 Stackelberg 博弈中的为公有云的报价和各资源的需求量, g_r^j 设为在时间段 τ 内的任务流对于资源的请求总量与边缘服务器目前资源的量的差, 当差值小于零时, 令 $g_r^j = 0$. 考虑 f_r^j 的取值, 此时 $U_1^j(f^j, g^j, p^j)$ 如公式 (34):

$$U_1^j(f^j, g^j, p^j) = \sum_r [x_r(q_r^c) \cdot \left(\frac{\varphi_r - f_r^j}{f_r^j - \varphi_r - \zeta E_r^{Cj}} + g_r^j + 1 \right) p_r^j - f_r^j \cdot \left(\frac{\varphi_r - f_r^j}{f_r^j - \varphi_r - \zeta E_r^{Cj}} + g_r^j + 1 \right)] \quad (34)$$

采用拉格朗日乘子法, 列出拉格朗日函数如公式 (35):

$$L(f_r^j, a) = \sum_r \left[x_r(q_r^c) \cdot \left(\frac{\varphi_r - f_r^j}{f_r^j - \varphi_r - \zeta E_r^{Cj}} + g_r^j + 1 \right) \cdot p_r^j - f_r^j \cdot \left(\frac{\varphi_r - f_r^j}{f_r^j - \varphi_r - \zeta E_r^{Cj}} + g_r^j + 1 \right) \right] - a \left(\sum_{r \in \Omega} f_r^j - M^j \right) \quad (35)$$

方程满足 Kuhn-Tucker 条件, 如公式 (36) 和公式 (37):

$$\frac{\partial L(f_r^j, a)}{\partial f_r^j} = 0 \quad (36)$$

$$\frac{\partial L(f_r^j, a)}{\partial a} = 0 \quad (37)$$

即满足公式 (38) 及公式 (39):

$$\frac{(\varphi_r)^2 + \varphi_r \cdot \zeta \cdot E_r^{Cj} + x_r(q_r^c) \cdot p_r^j \cdot E_r^{Cj} + f_r^j (f_r^j - 2(\varphi_r + \zeta \cdot E_r^{Cj}))}{(f_r^j - \varphi_r - \zeta \cdot E_r^{Cj})^2} - g_r^j - 1 + a = 0 \quad (38)$$

$$\left(\sum_{r \in \Omega} f_r^j - M^j \right) = 0 \quad (39)$$

为了方便表示, 令:

$$\Gamma = (\varphi_r)^2 + \varphi_r \cdot \zeta \cdot E_r^{Cj} + x_r(q_r^c) \cdot p_r^j \cdot E_r^{Cj} \quad (40)$$

$$\Theta = \varphi_r + \zeta E_r^{Cj} \quad (41)$$

当 $g_r^j - a \neq 0$ 时, 求得 f_r^j 及 a 满足公式 (42) 以及公式 (43):

$$f_r^j = \frac{\sqrt{(g_r^j - a)(-\Theta^2 - \Gamma)} + \Theta(g_r^j - a)}{g_r^j - a} \quad (42)$$

$$\sum_{r \in \Omega} f_r^j - M^j = \sum_{r \in \Omega} \frac{\sqrt{(g_r^j - a)(-\Theta^2 - \Gamma)} + \Theta(g_r^j - a)}{g_r^j - a} - M^j = 0 \quad (43)$$

解得, $a = \arg \sum_{r \in \Omega} \frac{\sqrt{(g_r^j - a)(-\Theta^2 - \Gamma)} + \Theta(g_r^j - a)}{g_r^j - a} - M^j$. a 可以根据梯度下降法或者二分法等方法求解. 那么最优策略 f_r^{j*} 解如公式 (44):

$$f_r^{j*} = \begin{cases} \frac{\sqrt{(g_r^j - a)(-\Theta^2 - \Gamma)} + \Theta(g_r^j - a)}{g_r^j - a}, & r \in \Omega' \\ 0, & r \in \Omega'' \end{cases} \quad (44)$$

基于文献 [32], 可知子迭代算法可以用来求解凸优化问题进而得到最优解. 当私有云策略得到最优解时, 没有任何一方可以通过单独修改自己的当前策略来提高自身的收益, 即该最优解为 Nash 均衡解. 且可以通过归纳假设证明私有云的最优策略在每一次迭代中均是固定的, 即算法 2 中的 Nash 均衡存在且唯一.

算法 3 为最优 Ω'' , Ω'' 选择策略算法, 其中 Ω' 中为最终向公有云报价竞争购买的资源类型, Ω'' 中为不向公有云购买的资源类型. 其主要由算法 3 求得. 在算法 3 中, 公有云会根据给定其他私有云支付的总价格来给出其分配资源策略 (第 2-4 行), 考虑到有限的预算, 边缘服务器需要在满足约束条件的资源中再次进行了一次筛选, 来获取有效的可用资源 (第 5-12 行). 求得了向公有云报价竞争购买的资源类型和不向公有云购买的资源类型.

综上, 由以上结论得出本文提出的双层 Stackelberg 博弈在公有云、私有云与用户之间存在 Nash 均衡, 且 Nash 均衡唯一.

3.3.4 与单层 Stackelberg 博弈比较

在单层 Stackelberg 博弈模型中^[28], 构造了一个边缘服务器为领导者, 每个移动设备为跟随者的两极博弈. 每个移动设备的成本函数和边缘服务器的收入定义: $k \in K$, 可以通过最小化成本函数 $U_k(l_k, f_k^c)$ 来确定要卸载的数据量 l_k 以及计算能力 f_k^c , 而边缘服务器可以选择一组 a 和 b 的最大化其收益函数 $Y(a, b)$, 从边缘服务器的角度来看给定定价函数 $P(\cdot)$, 即定价 a 和 b 的总收益 $Y(a, b)$ 可以写成公式 (45):

$$Y(a, b) = \sum_{k \in K} U_k^p = \sum_{k \in K} P(f_k^c) t_k^c \quad (45)$$

与单层 Stackelberg 博弈对比来看, 本文的双层 Stackelberg 博弈增加了公有云与私有云的构造, 不仅涵盖了单层 Stackelberg 博弈中边缘服务器与用户之间的交互, 还增加了公有云与私有云之间的交互, 所以可以得出结论, 双层 Stackelberg 博弈对比与单层 Stackelberg 是有优势的.

算法 3. 最优 Ω' , Ω'' 选择策略.

输入: $x_r(q_r^c), \phi_r, \xi, E_r^{Cj}, g_r^j$;

输出: Ω', Ω'' .

1. **For** r **in** Ω
 2. 计算 Γ, Θ ;
 3. 计算 α ;
 4. 根据公式 (44) 计算 f_r^j ;
 5. **If** $0 < f_r^j \leq \phi_r$
 6. $f_r^{j*} = 0$;
 7. $\Omega' \leftarrow r$;
 8. **Else**
 9. 根据公式 (44) 计算 f_r^{j*} ;
 10. $\Omega'' \leftarrow r$;
 11. **End if**
 12. **End for**
-

综上所述, 在固定私有云的策略时, 公有云的策略存在最优解 b_r^{j*} , 且在其余参与人的策略不变时, 公有云无法通过修改自己的策略使得收益提高; 私有云服务器 j 存在最优解 $f_r^{j*}, g_r^{j*}, D_r^{j*}$, 当其余参与竞争的私有云服务器以及公有云的策略不变时, 私有云服务器 j 也无法通过修改自己的策略使得收益提高; 同理用户也存在最优策略 a_r^{j*} , 且当其余参与人策略不变时, 无法通过改变自己的策略使得收益提高. 故可得出结论本文提出的双层 Stackelberg 博弈存在 Nash 均衡. 各参与人的最优解即为 Nash 均衡点, 由于参与人策略最优解的唯一性, 所以因为存在唯一的最优点, 即存在唯一 Nash 均衡点.

4 实验及评估

为验证本文提出的基于双层 Stackelberg 博弈的计算卸载方案的合理性、可行性和高效性, 通过模拟进行模拟实验, 实验首先研究算法 2 的收敛性, 来证明本方案具有合理性; 其次对私有云收益函数进行分析, 对比不同的 λ 下的收益, 并针对公有云收益函数进行分析, 对比不同报价策略下公有云收益, 证明基于双层 Stackelberg 博弈的可行性; 还对比了基于双层 Stackelberg 博弈和单层 Stackelberg 博弈的算法在不同条件下的私有云收益, 证明基于双层 Stackelberg 博弈的高效性, 且双层 Stackelberg 博弈相较于单层 Stackelberg 博弈有明显的优势.

4.1 实验配置

本实验在一台 i7-7700HQ CPU, 2.8 GHz 频率, 8 GB 内存的电脑上运行完成, 使用 Matlab 2014a 进行仿真模

拟, 具体参数及数据见表 2 实验参数表.

本方案适合任务量大时, 私有云无法单独处理, 而仅扩大私有云的规模则会导致在平时的资源浪费的环境, 这时则需要考虑到租借公有云, 而具体应用于某一环境时, 对于用户进行部署如图 6.

表 2 实验参数表

| 参数 | 值 | 参数说明 |
|----------------------|---------|-------------------------------------|
| y_r | [1,15) | 任务资源类型 r 的任务量 |
| E_r^{Cj} | 1 | 任务流的资源类型 r 的部分从服务器 j 运输到公有云上的能耗 |
| N | [1,10) | 任务数量 |
| ζ | 1 | 单位能耗的价格 |
| λ | [10,25) | 卸载概率参数 |
| $x_r(q_r^C)$ | 0.5 | 任务流的资源类型 r 的部分在公有云上的服务质量 |
| p_r^{bound} | [1,25) | 任务流的资源类型 r 的部分能接受的报价上限 |
| f_r^j | (4,5) | 服务器 j 对公有云的资源类型 r 的报价策略 |
| t_r^{bound} | [1,25) | 任务流的资源类型 r 的部分能接受的用时上限 |
| φ_r | 2 | 公有云的资源类型 r 出售给其他项目的平均收益 |
| p_r^j | [5,25) | 服务器 j 上资源类型 r 的功率 |
| g_r^j | 50 | 服务器 j 对公有云的资源类型 r 的资源量需求策略 |
| t_r^j | [15,25) | 任务流的资源类型 r 的部分在边缘服务器 j 上用时 |
| b_r^j | [40,50) | 公有云对服务器 j 的分配策略 |
| E_r^j | 2 | 任务流在服务器 j 上的总能耗 |

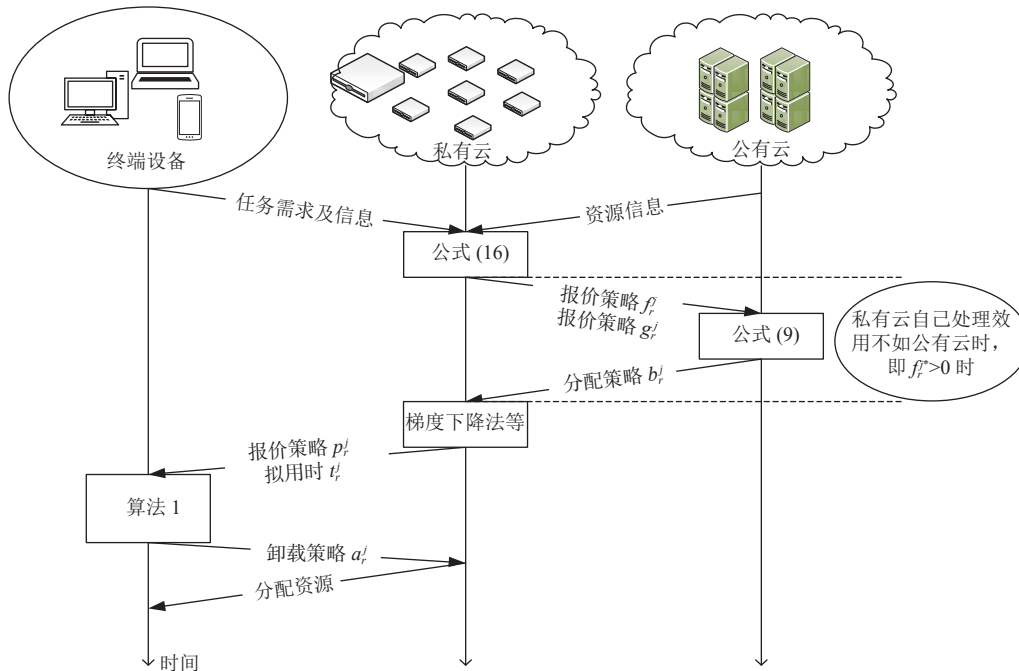


图 6 具体部署图

4.2 方案合理性与可行性验证

首先对算法 2 进行了分析, 如图 7 可以看出, 在经过 14 次迭代后, 私有云对于用户的报价 p_r^j 趋于稳定, 这也印证了前文中的理论分析. 即通过算法 2 的有限次迭代可以得到一个稳定的 Nash 均衡解, 证明了方案的合理性.

图 8 对于不同的卸载概率参数 λ 下的私有云服务器收益进行了讨论, 对 λ 进行了 0-50 的取值, 最终取了 10、15、20、25 这 4 个值更明显地看出 4 条曲线均为凸函数, 均存在最优解, 这也印证了前文的理论分析. 此外, 随着 λ 的增长, 私有云服务器的收益更慢达到峰值, 且峰值收益更高. 显然这主要是由于 λ 直接影响着 $x_r^j(q_r^j)$ 的取值, 且存在正相关.

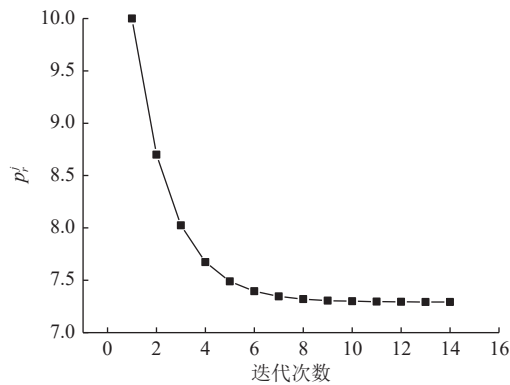


图 7 p_r^j 的收敛性

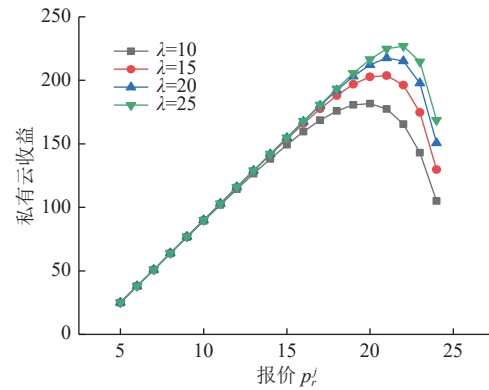


图 8 不同卸载概率参数 λ 下私有云的收益

图 9 主要针对公有云的效用函数进行讨论, 为了使对比图像更明显, 取了 4 个不同取值的 f_r^j , 可以看出, 公有云的收益随着分配策略的增加先增加后减少. 此外, 根据对于不同的报价策略下的收益曲线, 看出, 随着报价策略的增大, 公有云的收益也随之增大. 并且, f_r^j 越大的曲线到达峰值越迟, 即 f_r^j 越大的时候, 分配策略的最优解就越大, 即可以给予分配更多的资源.

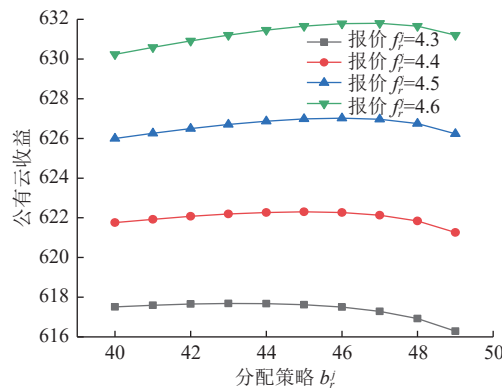


图 9 不同报价策略下公有云的收益

通过分析 4 个不同卸载概率参数 λ 值下的私有云服务器收益分析, 发现 λ 直接影响着 $x_r^j(q_r^j)$ 的值, 并且为正相关; 通过分析 4 个不同报价值下的公有云的收益, 发现报价策略越大则可分配更多的资源, 证明了方案的可行性.

4.3 方案高效性验证

图 10 对比了不同用时基于双层 Stackelberg 博弈和单层 Stackelberg 博弈的算法的私有云的收益, 从图 10 取了相对明显的用时可以看出, 基于双层 Stackelberg 博弈和单层 Stackelberg 博弈的私有云的收益函数趋势相差不

多,但是由于双层 Stackelberg 博弈考虑到了公有云,可以获得更多的任务,即可以获得更多的收益.此外,用时即任务来了以后,边缘服务器来处理时可用的时长,任务的量的不同则模拟的用时不同,当用时趋于 t_r^{bound} 时,基于单层 Stackelberg 博弈下降趋势更大.

图 11 和图 12,对比了不同报价基于双层 Stackelberg 博弈和单层 Stackelberg 博弈的算法的私有云的收益,图 11 为报价较小时,基于双层 Stackelberg 博弈和单层 Stackelberg 博弈的私有云的收益函数的比较,发现私有云收益差不多.而图 12 取了相对明显的报价,发现当报价增长时,基于单层 Stackelberg 博弈上涨趋势更小,这就说明了基于双层 Stackelberg 博弈相较单层对于私有云来说,可以带来更多的收益.

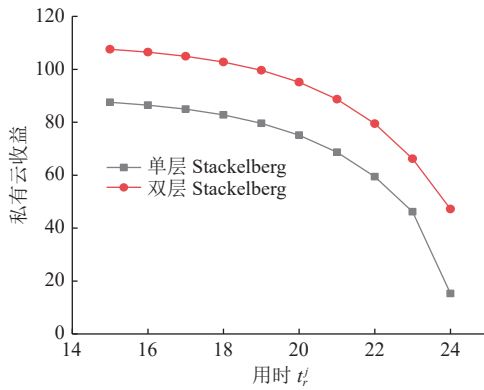


图 10 不同用时 t_r^j 下的私有云收益

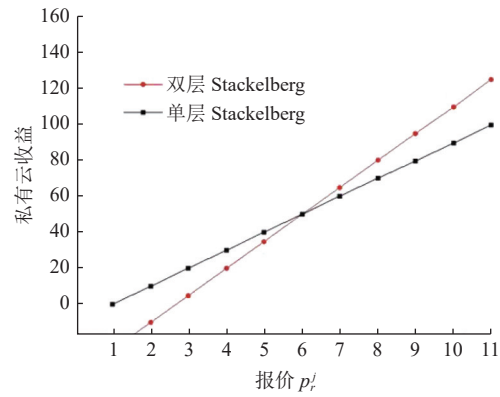


图 11 较小报价 p_r^j 下的私有云收益

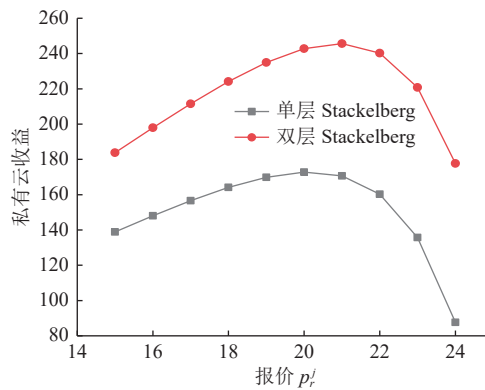


图 12 较大报价 p_r^j 下的私有云收益

算法 1 即用户的最优策略算法是对用户的最优策略进行求解,其复杂度为 $O(\log m)$; 算法 2 即私有云的最优策略算法是对私有云的最优策略进行求解,其复杂度为 $O(\log m)$; 算法 3 即最优 Ω' , Ω'' 选择策略算法是对公有云的最优策略进行求解,其复杂度为 $O(m)$.

Zhou 等人^[30]利用非合作博弈解决部分卸载问题提出了 NPCA 算法,并也做了复杂度分析及纳什均衡的存在性.从表 3 看出,Zhou 等人^[30]提出的部分计算卸载算法的复杂度为 $O(SL \log L)$, S 为分组的数目、 L 为每组的长度.与本文所提出的算法复杂度对比,比本文的算法的复杂度高. Liang 等人^[28]采用了 Stackelberg 博弈来研究基于定价策略的 MEC 计算卸载问题,通过实验求解了非凸的问题,验证 Stackelberg 博弈算法效果要比线性策略更好.而本文的计算卸载方案是基于双层 Stackelberg 博弈的,通过上述实验证明了双层 Stackelberg 博弈要比单层 Stackelberg 博弈要好,所以我们的算法要比 Liang 等人提出的算法效果好,证明了基于双层 Stackelberg 博弈的计算卸载方案的优势.

总体来说,通过以上实验,验证了基于双层 Stackelberg 博弈的计算卸载方案的合理性、可行性和高效性.首

先发现在经过 14 次迭代后, 私有云对于用户的报价将趋于稳定, 值为 7.25, 即私有云的最优策略算法有良好的收敛性, 证明了方案的合理性; 其次对比不同的卸载概率参数 λ 下的私有云服务器收益, 发现其 4 条曲线均为凸函数, 并且对比不同报价策略下公有云的收益, 发现报价策略越大则可分配更多的资源, 证明了 Nash 均衡解的可行性, 证明基于双层 Stackelberg 博弈的可行性; 最后通过对比单层 Stackelberg 博弈和双层 Stackelberg 博弈, 对于不同用时下的私有云服务器收益与不同的报价下的公有云服务器收益进行了分析, 证明了相较于单层 Stackelberg 博弈, 基于双层 Stackelberg 博弈的计算卸载模型更具有优势, 且证明了基于双层 Stackelberg 博弈的计算卸载模型的高效性。

表 3 时间复杂度对比

| 算法名称 | 复杂度 |
|----------------------------------|----------------|
| 算法1. 用户的最优策略 | $O(\log m)$ |
| 算法2. 私有云的最优策略 | $O(\log m)$ |
| 算法3. 最优 Ω', Ω'' 选择策略 | $O(m)$ |
| 部分计算卸载算法 ^[30] | $O(SL \log L)$ |

5 总结

针对 MEC 的计算卸载问题, 本文不同于传统的计算卸载方案, 提出了一种新的基于双层 Stackelberg 博弈的计算卸载方案, 该方案考虑了公有云与私有云之间的关系, 将公有云作为私有云资源的补充, 可以缓解由于私有云资源局限性带来的算力不足问题; 本文采用了双层 Stackelberg 博弈建立模型, 该模型采用了部分卸载的方式, 可以分布式地解决 MEC 的计算卸载问题, 对于公有云、私有云以及用户的策略和收益进行了分析, 最终求出了各参与人的最优策略, 证明了双层博弈纳什均衡解的存在性及唯一性。对比了基于双层 Stackelberg 博弈与基于单层 Stackelberg 博弈的理论, 证明了本方案更有优势; 此外进行了实验, 分析了求解算法的收敛性以及公有云和私有云的效用函数的变化以及受参数变动的的影响, 证明了解的收敛性以及设计的合理性与可行性。最后还对比了基于单层 Stackelberg 博弈和双层 Stackelberg 博弈的卸载模型, 证明了基于双层 Stackelberg 博弈的计算卸载模型的高效性。

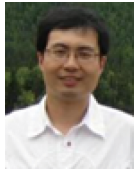
本文把边缘计算中的中心云和边缘云抽象为一个个体私有云, 而没有涉及各边缘云间的计算卸载方案, 在未来将考虑将多层 Stackelberg 博弈等方法到边缘云和私有云间的计算卸载方案中。论文所提出的基于双层 Stackelberg 博弈的计算卸载方案现在还没有进行实际的应用验证, 这也是未来工作之一。

References:

- [1] Hao YX, Jiang YG, Hossain MS, Alhamid MF, Amin SU. Learning for smart edge: Cognitive learning-based computation offloading. *Mobile Networks and Applications*, 2020, 25(3): 1016–1022. [doi: 10.1007/s11036-018-1119-7]
- [2] Ndikumana A, Tran NH, Ho TH, Han Z, Saad W, Niyato D, Hong CS. Joint communication, computation, caching, and control in big data multi-access edge computing. *IEEE Trans. on Mobile Computing*, 2020, 19(6): 1359–1374. [doi: 10.1109/TMC.2019.2908403]
- [3] Ren JK, Yu GD, Cai YL, He YH, Qu FZ. Partial offloading for latency minimization in mobile-edge computing. In: *Proc. of the 2017 IEEE Global Communications Conf. Singapore: IEEE*, 2017. 1–6. [doi: 10.1109/GLOCOM.2017.8254550]
- [4] Mahmood A, Ahmed A, Naeem M, Hong Y. Partial offloading in energy harvested mobile edge computing: A direct search approach. *IEEE Access*, 2020, 8: 36757–36763. [doi: 10.1109/ACCESS.2020.2974809]
- [5] Tuong VD, Truong TP, Nguyen TV, Noh W, Cho S. Partial computation offloading in NOMA-assisted mobile-edge computing systems using deep reinforcement learning. *IEEE Internet of Things Journal*, 2021, 8(17): 13196–13208. [doi: 10.1109/JIOT.2021.3064995]
- [6] Zhao HL, Du W, Liu W, Lei T, Lei QW. QoE aware and cell capacity enhanced computation offloading for multi-server mobile edge computing systems with energy harvesting devices. In: *Proc. of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation. Guangzhou: IEEE*, 2018. 671–678. [doi: 10.1109/SmartWorld.2018.00133]
- [7] Huang BB, Li ZJ, Tang P, Wang SG, Zhao J, Hu HY, Li WQ, Chang V. Security modeling and efficient computation offloading for

- service workflow in mobile edge computing. *Future Generation Computer Systems*, 2019, 97: 755–774. [doi: [10.1016/j.future.2019.03.011](https://doi.org/10.1016/j.future.2019.03.011)]
- [8] Goudarzi M, Zamani M, Haghghat AT. A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing. *Int'l Journal of Communication Systems*, 2017, 30(10): e3241. [doi: [10.1002/dac.3241](https://doi.org/10.1002/dac.3241)]
- [9] Liu JH, Zhang Q. Code-partitioning offloading schemes in mobile edge computing for augmented reality. *IEEE Access*, 2019, 7: 11222–11236. [doi: [10.1109/ACCESS.2019.2891113](https://doi.org/10.1109/ACCESS.2019.2891113)]
- [10] Huynh LNT, Pham QV, Pham XQ, Nguyen TDT, Hossain D, Huh EN. Efficient computation offloading in multi-tier multi-access edge computing systems: A particle swarm optimization approach. *Applied Sciences*, 2020, 10(1): 203. [doi: [10.3390/app10010203](https://doi.org/10.3390/app10010203)]
- [11] Wang QY, Mao YC, Wang YC, Wang LB. Computation tasks offloading scheme based on multi-cloudlet collaboration for edge computing. In: *Proc. of the 7th Int'l Conf on Advanced Cloud and Big Data*. Suzhou: IEEE, 2019. 339–344. [doi: [10.1109/CBD.2019.00067](https://doi.org/10.1109/CBD.2019.00067)]
- [12] Shakarami A, Ghobaei-Arani M, Shahidinejad A. A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. *Computer Networks*, 2020, 182: 107496. [doi: [10.1016/j.comnet.2020.107496](https://doi.org/10.1016/j.comnet.2020.107496)]
- [13] Shin KY, Jeong HJ, Moon SM. Enhanced partitioning of DNN layers for uploading from mobile devices to edge servers. In: *Proc. of the 3rd Int'l Workshop on Deep Learning for Mobile Systems and Applications*. New York: ACM, 2019. 35–40. [doi: [10.1145/3325413.3329788](https://doi.org/10.1145/3325413.3329788)]
- [14] Jeong HJ, Lee HJ, Shin CH, Moon SM. IONN: Incremental offloading of neural network computations from mobile devices to edge servers. In: *Proc. of ACM Symp. on Cloud Computing*. Carlsbad: ACM, 2018. 401–411. [doi: [10.1145/3267809.3267828](https://doi.org/10.1145/3267809.3267828)]
- [15] Huang L, Feng X, Feng AQ, Huang YP, Qian LP. Distributed deep learning-based offloading for mobile edge computing networks. *Mobile Networks and Applications*, 2018. [doi: [10.1007/s11036-018-1177-x](https://doi.org/10.1007/s11036-018-1177-x)]
- [16] Huang L, Feng X, Zhang C, Qian LP, Wu Y. Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing. *Digital Communications and Networks*, 2019, 5(1): 10–17. [doi: [10.1016/j.dcan.2018.10.003](https://doi.org/10.1016/j.dcan.2018.10.003)]
- [17] Chen XF, Zhang HG, Wu CLMG, Mao SW, Ji YS, Bennis M. Performance optimization in mobile-edge computing via deep reinforcement learning. In: *Proc. of the 88th IEEE Vehicular Technology Conf*. Chicago: IEEE, 2018. 1–6. [doi: [10.1109/VTCSFall.2018.8690980](https://doi.org/10.1109/VTCSFall.2018.8690980)]
- [18] Yang TY, Hu YL, Gursoy MC, Schmeink A, Mathar R. Deep reinforcement learning based resource allocation in low latency edge computing networks. In: *Proc. of the 15th Int'l Symp. on Wireless Communication Systems*. Lisbon: IEEE, 2018. 1–5. [doi: [10.1109/ISWCS.2018.8491089](https://doi.org/10.1109/ISWCS.2018.8491089)]
- [19] Lu HF, Gu CH, Luo F, Ding WC, Liu XP. Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning. *Future Generation Computer Systems*, 2020, 102: 847–861. [doi: [10.1016/j.future.2019.07.019](https://doi.org/10.1016/j.future.2019.07.019)]
- [20] Tan LT, Hu RQ. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning. *IEEE Trans. on Vehicular Technology*, 2018, 67(11): 10190–10203. [doi: [10.1109/TVT.2018.2867191](https://doi.org/10.1109/TVT.2018.2867191)]
- [21] Chen Z, Wang XD. Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach. *EURASIP Journal on Wireless Communications and Networking*, 2020, 2020(1): 188. [doi: [10.1186/s13638-020-01801-6](https://doi.org/10.1186/s13638-020-01801-6)]
- [22] Zeng DZ, Gu L, Pan SL, Cai JJ, Guo S. Resource management at the network edge: A deep reinforcement learning approach. *IEEE Network*, 2019, 33(3): 26–33. [doi: [10.1109/MNET.2019.1800386](https://doi.org/10.1109/MNET.2019.1800386)]
- [23] Verma T, Kumar A. *Fuzzy Solution Concepts for Non-cooperative Games*. Cham: Springer, 2020. [doi: [10.1007/978-3-030-16162-0](https://doi.org/10.1007/978-3-030-16162-0)]
- [24] Mukherjee A, Deb P, De D, Buyya R. C2OF2N: A low power cooperative code offloading method for femtolet-based fog network. *The Journal of Supercomputing*, 2018, 74(6): 2412–2448. [doi: [10.1007/s11227-018-2269-x](https://doi.org/10.1007/s11227-018-2269-x)]
- [25] Guo HZ, Liu JJ, Qin HL. Collaborative mobile edge computation offloading for IoT over fiber-wireless networks. *IEEE Network*, 2018, 32(1): 66–71. [doi: [10.1109/MNET.2018.1700139](https://doi.org/10.1109/MNET.2018.1700139)]
- [26] Chen X. Decentralized computation offloading game for mobile cloud computing. *IEEE Trans. on Parallel and Distributed Systems*, 2015, 26(4): 974–983. [doi: [10.1109/TPDS.2014.2316834](https://doi.org/10.1109/TPDS.2014.2316834)]
- [27] Jošilo S, Dán G. Wireless and computing resource allocation for selfish computation offloading in edge computing. In: *Proc. of the 2019 IEEE Conf. on Computer Communications*. Paris: IEEE, 2019. 2467–2475. [doi: [10.1109/INFOCOM.2019.8737480](https://doi.org/10.1109/INFOCOM.2019.8737480)]
- [28] Liang BZ, Fan RF, Hu H, Zhang Y, Zhang N, Anpalagan A. Nonlinear pricing based distributed offloading in multi-user mobile edge computing. *IEEE Trans. on Vehicular Technology*, 2021, 70(1): 1077–1082. [doi: [10.1109/TVT.2020.3045473](https://doi.org/10.1109/TVT.2020.3045473)]
- [29] Lei Y, Zheng WB, Ma Y, Xia YN, Xia Q. A novel probabilistic-performance-aware and evolutionary game-theoretic approach to task offloading in the hybrid cloud-edge environment. In: *Proc. of the 16th Int'l Conf. on Collaborative Computing: Networking, Applications and Worksharing*. Shanghai: Springer, 2020. 255–270. [doi: [10.1007/978-3-030-67537-0_16](https://doi.org/10.1007/978-3-030-67537-0_16)]

- [30] Zhou SC, Jadoon W. The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment. *Computer Networks*, 2020, 178: 107334. [doi: [10.1016/j.comnet.2020.107334](https://doi.org/10.1016/j.comnet.2020.107334)]
- [31] Gao L, Moh M. Joint Computation Offloading and Prioritized Scheduling in Mobile Edge Computing. In: *Proc. of the 2018 Int'l Conf. on High Performance Computing & Simulation (HPCS)*. IEEE, 2018. 1000–1007. [doi: [10.1109/HPCS.2018.00157](https://doi.org/10.1109/HPCS.2018.00157)]
- [32] Xiao Y, Bi GA, Niyato D. A simple distributed power control algorithm for cognitive radio networks. *IEEE Trans. on Wireless Communications*, 2011, 10(11): 3594–3600. [doi: [10.1109/TWC.2011.090611.102049](https://doi.org/10.1109/TWC.2011.090611.102049)]



孙伟峰(1978—), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为边缘计算, 无线网络协议优化, 基于 SDN 的跨层智能算法设计.



江贺(1980—), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为基于搜索的软件工程, 软件仓库挖掘.



张渊樾(1998—), 男, 硕士, 主要研究领域为边缘计算中的任务卸载与资源分配.



秦一星(1999—), 女, 硕士生, 主要研究领域为边缘计算.