

移动边缘网络下服务缓存与资源分配联合优化策略

龙隆¹, 刘子辰¹, 陆在旺^{1,2}, 张玉成^{1,2}, 李蕾^{1,2}

(1. 中国科学院计算技术研究所, 北京 100190;

2. 中国科学院大学计算技术研究所, 北京 100049)

摘 要: 针对边缘服务器的计算与存储资源有限, 过多的任务卸载将使其计算能力与负载能力不匹配, 从而导致任务处理时延增加的问题, 研究了基于多用户、边缘服务器以及云服务器组成的三层网络架构下的任务卸载与存储资源联合优化问题以降低系统整体时延。由于该问题为混合整数非线性问题, 因此提出了一种服务缓存与资源分配联合优化策略。首先, 将原问题的连续与离散变量进行解耦为 2 个子问题, 即服务缓存决策问题以及计算资源与通信资源联合优化问题。然后, 通过重构线性化方法、松弛法以及凸优化方法对 2 个子问题进行交替优化迭代获得近似最优解。仿真结果表明, 所提策略在低复杂度情况下能够获得近似最优解, 并且与其他策略相比时延降低 10% 左右。

关键词: 移动边缘计算; 多级网络; 服务缓存; 资源分配

中图分类号: TN929.52

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2023020

Joint optimization strategy of service cache and resource allocation in mobile edge network

LONG Long¹, LIU Zichen¹, LU Zaiwang^{1,2}, ZHANG Yucheng^{1,2}, LI Lei^{1,2}

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

2. Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing 100190, China

Abstract: Aiming at the problem that computing and storage resources of edge node in a mobile edge computing (MEC) system were limited, and excessive task offloading would cause the mismatch between the computing capacity and the load capacity of the edge server, resulting in the increase of task processing delay, the joint optimization of task offloading and storage resources under the three-layer network architecture composed of multi-user, edge server and cloud server was studied to reduce the overall system delay. For the problem was a mixed integer nonlinear problem, a joint optimization strategy of service caching and resource allocation was proposed. First, the continuous and discrete variables of the original problem were decoupled into two sub-problems, namely, the service cache decision problem and the joint optimization problem of computing resources and communication resources. Then, the linear reconstruction, relaxation method and convex optimization method were used to alternately optimize the two sub-problems to obtain the near-optimal solution. Simulation results demonstrate that the proposed strategy can obtain a near-optimal performance with low complexity, and can reduce up to 10% of the task duration compared with other strategies.

Keywords: mobile edge computing, multilevel network, service cache, resource allocation

收稿日期: 2022-09-07; 修回日期: 2022-12-05

通信作者: 刘子辰, liuzichen@ict.ac.cn

基金项目: 国家重点研发计划基金资助项目 (No.2022YFD2001004)

Foundation Item: The National Key Research and Development Program of China (No.2022YFD2001004)

0 引言

随着移动通信与物联网的高速发展,许多计算密集和时延敏感应用(如无人驾驶、图像识别以及自然语言处理等)产生,这类具有低时延高可靠特点的应用对终端设备的计算能力提出较高要求^[1]。由于计算能力有限的终端设备在处理这类应用时将产生较高的任务时延,因此如何降低应用处理时延是目前亟待解决的关键问题之一。移动云计算(MCC, mobile cloud computing)技术的出现很好地解决了终端计算能力受限的问题。MCC的目标是将云端丰富的计算资源扩展至资源受限的终端设备,从而增强终端设备潜在的计算能力^[2-4]。终端设备与云服务器间较远的通信距离所产生的网络时延使单纯依靠MCC无法满足时延敏感型应用程序的时延要求。为了克服以上问题,并对MCC技术进行补充,移动边缘计算(MEC, mobile edge computing)作为一种新的技术被提出^[5-6]。移动边缘计算主要由基站及其边缘服务器组成。当终端设备执行计算任务时,由基站与服务器组成的边缘节点可以就近为这些终端提供计算、通信以及数据存储等服务,从而降低任务执行时延、提高系统性能。目前,基于MEC网络的任务卸载和资源分配等问题已有相关研究。文献[7-9]在不同应用场景下以最大化能效为目标,对系统频谱资源和功耗等进行联合优化。文献[10-11]分别以最小化能耗、时延为目标,对计算与频谱资源进行联合优化。然而在实际应用中,边缘服务器的计算资源与存储资源是有限的,过多的计算请求必然为边缘服务器带来较高的计算与存储压力,进而影响系统性能。MCC与MEC融合网络下的资源分配与卸载方案为解决以上问题提供了新的解决思路。在MCC与MEC融合的网络架构中,终端设备进行任务请求时,既可以在本地执行,也可以在边缘服务器或云端执行。因此,该网络架构可以为终端提供更高效且更灵活的卸载服务,从而满足不同的服务需求。目前已有相关研究表明,在三级网络架构下通过卸载决策与资源分配联合优化的方式可以提升网络性能^[12-13]。在MEC与MCC融合的集中式网络下,文献[12]对卸载决策与资源进行联合优化,以降低所有终端的任务时延;文献[13]以最大化系统收益为目标,对任务卸载与资源分配进行联合优化设计。

上述研究从不同方面解决任务卸载与资源分

配联合优化问题,但是这些研究忽略了任务卸载与服务缓存耦合性对系统性能的影响。这里的服务缓存是指将与计算任务相关的数据库、应用程序、函数库以及相关代码等放在边缘服务中,从而支持不同类型的应用程序。从任务卸载角度可以将现有MEC研究分为两大类,即计算任务完全卸载与计算任务部分卸载。完全卸载模型适用于高度集成或相对简单的任务,如自然语言处理、导航等任务,这些任务不能被分割,并且只能在移动设备上执行或者作为一个整体由边缘服务器执行。部分卸载模型适用于可分割的任务,如图像识别、增强现实(AR, augmented reality)应用等,这类应用的任务可以被分割为多个子任务通过并行或串行的方式执行。与完全卸载模型相比,基于部分卸载模型的资源分配更加合理,资源利用率也更高。目前,关于服务缓存与资源分配已有相关研究。文献[14]以最小化供应商成本为目标,提出了一种整数线性规划和随机搜索算法,对边缘节点间的服务缓存资源进行优化。文献[15]以最小化服务缓存总成本为目标,对边缘服务器的位置 and 用户服务需求进行联合优化。文献[16]以最小化时延与成本为目标,对边缘服务器的内容缓存进行协同优化。尽管以上研究通过任务卸载与服务缓存优化使系统性能得到提升,但都是基于任务不可分的情况进行的资源分配优化,而对任务可分的情况不适用。

大多数复杂的应用程序由多个不同的执行组件组成,如执行实时面部识别任务时需要将其分为4个子任务分别处理,即人脸检测、图像处理与特征提取、人脸识别。这些子任务所需的计算资源、存储资源以及带宽资源各不相同。因此,如何基于部分任务卸载和服务缓存联合优化是目前亟待解决问题。文献[17]通过构建服务链模型,在存储资源以及计算资源约束下以最小化网络响应时延为目标,通过开放马尔可夫排队网络模型构建边缘节点服务调度模型,并对缓存决策与计算资源联合优化,但在任务卸载过程中忽略终端设备以及云服务器资源等可用资源。文献[18]基于部分卸载模型,在计算资源约束下对服务缓存与任务卸载进行联合优化研究,但其假设带宽资源均匀分配,导致计算数据量较大的任务得到较少的带宽资源而产生较高时延,计算数据量较少的任务得到

较多的带宽资源造成资源浪费。另外,文献[18]并未对边缘节点的存储空间进行约束,这意味着在该研究中边缘服务端拥有无限大的存储空间,可以缓存所有任务执行所需的相关数据,这在实际应用场景下显然不合理。

综上所述,本文将在终端、边缘服务以及云构成的三层网络架构下,综合考虑无线带宽资源、计算资源以及存储资源等因素。通过计算、通信、存储融合的思想,以最小化所有终端设备的任务时延为目标,综合考虑存储资源、计算资源、卸载决策以及任务卸载比例等对任务时延的影响,提出一种服务缓存与资源分配的联合优化算法。具体创新点介绍如下。

1) 构建了一个由多用户、边缘服务器以及云服务器组成的三层网络架构,每个用户的计算任务可以通过部分卸载的方式将任务分别卸载至本地、边缘服务器或云服务器。引入了最小化终端任务时延模型,考虑了用户计算资源、边缘服务的计算资源、存储资源以及带宽资源等因素对时延的影响。

2) 提出了服务缓存与资源分配联合(JSCRA, joint service caching and resource allocation)优化策略。首先,将原问题的连续变量与离散变量进行解耦,变为2个子问题,即服务

缓存决策问题、计算资源与通信资源联合优化问题。其次,基于重构线性化技术(RLT, reformulation linearization technique)、松弛以及凸优化方法对2个子问题进行交替迭代优化。最后,获得问题次优解。

3) 仿真结果表明,与随机缓存和资源分配(RCRA, random caching and resource allocation)策略相比,所提策略在收敛速度和性能表现上都有明显优势,可以获得更低的任务时延。

1 系统模型

1.1 集中式移动边缘网络模型

集中式移动边缘网络如图1所示。从图1可以看出,集中式移动边缘网络由云服务器、单一服务节点(由边缘服务器与基站组成)以及 L 个具有计算与通信能力的终端组成。这里的终端泛指具有计算能力的终端设备,如手机、平板电脑、车辆等。终端个数表示为 $\mathcal{L} = \{1, 2, \dots, L\}$, $i \in \mathcal{L}$ 。云服务器与服务节点通过光纤进行有线连接,为终端提供稳定的通信、计算以及存储服务。这里的云服务器主要由云服务器集群组成,因此拥有丰富的计算资源与存储资源。

在图1所示场景中,服务节点不仅可以为终端提供通信、计算以及存储资源,而且扮演着协调器

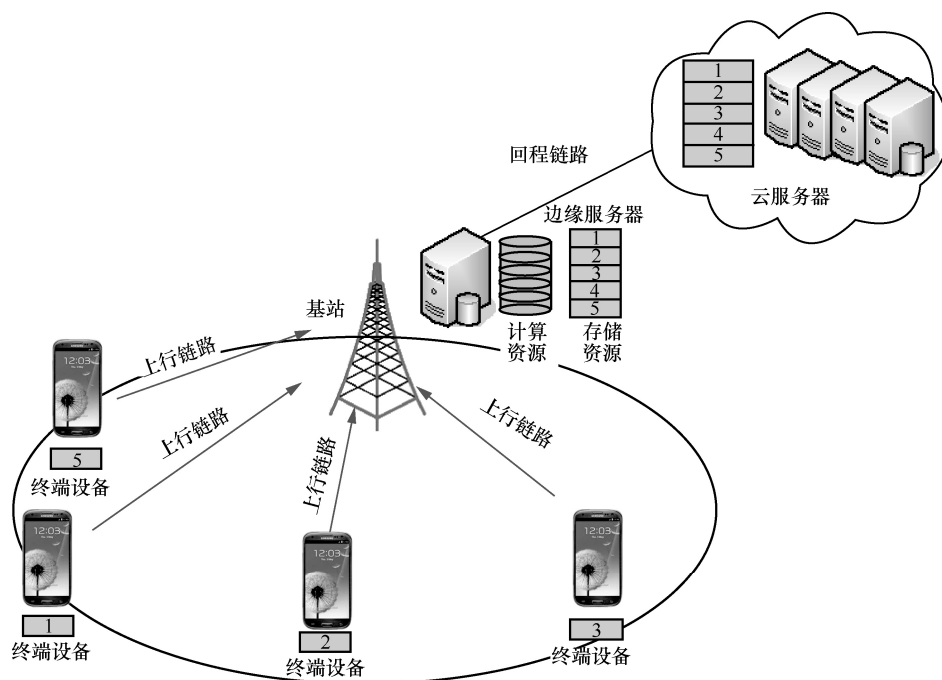


图1 集中式移动边缘网络

的角色，即根据终端的任务请求提供存储决策以及计算决策。这里假设边缘服务器存储计算服务的最大存储容量为 C ，最大计算能力为 F 。计算服务数据主要由数据库、数据字典以及相关的应用代码组成，其中每一种应用代表一个类型的计算任务。此外，由于存储是一个长期的过程，因此需要服务缓存进行定时更新。本文认为系统在离散的时间段中运行，每个时间段的持续时间与服务缓存和任务分配决策更新的时间尺度相匹配。本文对其中一个时间段进行研究，忽略其他时间段的索引^[18]。在一个时间段的开始阶段，每个终端随机请求一个计算任务，其中一类计算任务对应一类计算服务。这里假设存在计算服务 $\mathcal{K} = \{1, 2, \dots, K\}$ ， $w_{A_i} = \{c_{A_i}^{\text{input}}, s_i\}$ 表示用户 i 请求计算服务 A_i 相应的任务， $A_i \in \mathcal{K}$ ，其中， $c_{A_i}^{\text{input}}$ 表示终端 i 执行服务 A_i 相关任务的输入数据大小， s_i 表示执行 1 bit 数据所需循环数。为了便于分析，本文做出如下合理假设。

1) 服务节点已知信道状态信息 (CSI, channel state information)、每个任务输入数据的大小以及执行任务所需的循环数。基于以上假设，可以得到优化的卸载决策以及资源分配^[19]。

2) 每种应用的计算结果数据都远小于计算输入数据，因此忽略计算结果的回程链路传输时延^[20]。

1.2 通信模型

本文采用正交频分复用的方式将频谱资源分配给终端设备，频谱的总带宽为 B 。由于每个终端设备将计算任务卸载至服务节点时会占用上行频谱资源，因此终端 i 执行服务 A_i 相关任务的上行传输速率可以表示为^[12]

$$r_{A_i}^{\text{up}} = \theta_{A_i} B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right) \quad (1)$$

其中， θ_{A_i} 为终端上传数据所占上行带宽百分比， $P_{i,s}$ 为终端的发送功率， $h_{i,B}$ 为基站与终端间的信道衰落系数， d 为终端与基站的距离， r 为路径损耗， σ^2 为信道的噪声功率。

1.3 服务缓存模型

若边缘服务器存储计算服务，则意味着服务节点可以处理相关的计算任务，否则将任务交由云服务器执行。由于边缘服务器的存储空间有限，故无法满足所有终端设备的计算请求，因此边缘服务器需要对所缓存的服务数据进行决策。这里

将服务端的存储决策表示为 $X = \{x_{A_1}, x_{A_2}, \dots, x_{A_i}\}$ ， $x_{A_i} \in \{0, 1\}$ 且 $A_i \in \mathcal{K}$ ， $i \in \mathcal{L}$ ； x_{A_i} 表示服务器是否缓存终端设备请求的 A_i ， $x_{A_i} = 1$ 表示服务器缓存计算服务 A_i ， $x_{A_i} = 0$ 表示服务器没有缓存相应的计算服务 A_i 。每个服务 A_i 所需存储大小为 c_{A_i} ，而边缘服务器缓存决策受到存储容量的限制，即 $\sum_{i \in \mathcal{L}} x_{A_i} c_{A_i} \leq C, A_i \in \mathcal{K}$ 。

1.4 计算任务卸载模型

任务部分卸载意味着计算任务可以分解成多个大小不同的子任务。本文将基于卸载模型（即向代码分区的应用，如图像识别，这类应用分为多个模块，其中一个模块的输出结果为下一个模块的输入）进行研究，接下来本文将在部分卸载模型下，对任务执行时延模型进行描述。

1) 当计算服务 A_i 存储在边缘服务器时，任务可以在边缘服务器执行，此时任务执行时延分析如下。

如果部分任务在本地卸载，任务卸载的百分比用 β_{A_i} 表示，任务的本地卸载量用 $c_{A_i}^{\text{local}}$ 表示，即 $c_{A_i}^{\text{local}} = c_{A_i}^{\text{input}} \beta_{A_i}$ ， $0 \leq \beta_{A_i} \leq 1$ ， $\forall i \in \mathcal{L}$ ， $A_i \in \mathcal{K}$ ，本地的计算能力为 f_i^{local} ，则任务的本地计算时延表示为

$$t_{A_i}^{\text{local}} = \frac{c_{A_i}^{\text{local}} s_i}{f_i^{\text{local}}} \quad (2)$$

如果部分任务在服务节点卸载，任务卸载至边缘服务器时，任务卸载百分比表示为 $1 - \beta_{A_i}$ ，相应的任务卸载至边缘服务器的数据大小表示为 $c_{A_i}^{\text{mec}} = c_{A_i}^{\text{input}} (1 - \beta_{A_i})$ 。分配给终端设备的计算资源表示为 f_i^{mec} ，任务在边缘服务器的执行时延表示为

$$t_{A_i}^{\text{mec}} = \frac{c_{A_i}^{\text{mec}} s_i}{f_i^{\text{mec}}} \quad (3)$$

当任务卸载至边缘服务器时，终端设备将占用上行链路对数据进行传输，此时上行链路的传输时延表示为

$$t_{A_i}^{\text{up}} = \frac{c_{A_i}^{\text{mec}}}{r_{A_i}^{\text{up}}} \quad (4)$$

综上所述，当计算服务存储在服务中心时，任务执行总时延表示为

$$U_{A_i}^{\text{local_mec}} = t_{A_i}^{\text{local}} + t_{A_i}^{\text{up}} + t_{A_i}^{\text{mec}} \quad (5)$$

2) 当计算服务 A_i 未存储在边缘服务器时, 计算任务将在本地以及云端进行部分卸载。

如果部分任务卸载至本地, 任务卸载的百分比用 α_{A_i} 表示, 任务的本地卸载量用 $c_{A_i}^{\text{local_nocach}}$ 表示, 即 $c_{A_i}^{\text{local_nocach}} = c_{A_i}^{\text{input}} \alpha_{A_i}$, $0 \leq \alpha_{A_i} \leq 1$, $\forall i \in \mathcal{L}$, $\forall k \in \mathcal{K}$ 本地的计算能力为 f_i^{local} , 因此任务的本地计算时延表示为

$$t_{A_i}^{\text{local_nocach}} = \frac{c_{A_i}^{\text{local_nocach}} s_i}{f_i^{\text{local}}} \quad (6)$$

如果任务卸载至云服务器, 卸载百分比表示为 $1 - \alpha_{A_i}$, 其中, 任务卸载至边缘服务器的数据大小表示为 $c_{A_i}^{\text{cloud}} = c_{A_i}^{\text{input}} (1 - \alpha_{A_i})$ 。基于前文假设, 云服务器具有丰富的计算资源, 所以任务执行时延只与传输时延有关, 即上行链路传输时延以及回程链路传输时延。因此, 任务在云服务器执行时其传输时延表示为

$$t_{A_i}^{\text{cloud}} = \frac{c_{A_i}^{\text{cloud}}}{R} + \frac{c_{A_i}^{\text{cloud}}}{r_{A_i}^{\text{up}}} \quad (7)$$

其中, R 为边缘服务与云服务器之间的传输速率。因此, 计算服务存储在云服务器时任务执行总时延表示为

$$U_{A_i}^{\text{local_cloud}} = t_{A_i}^{\text{local_nocach}} + t_{A_i}^{\text{cloud}} \quad (8)$$

2 问题的形式化描述

由于存储是一个长期统计的过程, 因此其在短期内变化很小, 但计算与通信资源是实时的, 需要根据不同用户的任务需求实时变化。基于以上特点, 本文考虑在当前时间段内对计算、通信以及存储资源进行联合优化。本文的主要目标是在计算、通信、存储资源约束条件下, 对服务缓存决策 x 、上行频谱资源 θ 、任务部分卸载百分比 α 和 β , 以及服务器计算资源 f 联合优化, 以最小化所有终端设备执行任务总时延。优化目标可以表示为

$$\begin{aligned} \text{P0: } \min_{x, \alpha, \theta, \beta, f} & \sum_{i=1}^L x_{A_i} U_{A_i}^{\text{local_mec}} + (1 - x_{A_i}) U_{A_i}^{\text{local_cloud}} \\ \text{s.t. } & \text{C1: } \sum_{i \in \mathcal{L}} x_{A_i} f_i^{\text{mec}} \leq F \\ & \text{C2: } \sum_{i \in \mathcal{L}} \theta_{A_i} \leq 1 \\ & \text{C3: } 0 \leq \alpha_{A_i} \leq 1, 0 \leq \beta_{A_i} \leq 1, \quad \forall i \in \mathcal{L} \\ & \text{C4: } x_{A_i} \in \{0, 1\}, \quad \forall i \in \mathcal{L} \\ & \text{C5: } \sum_{i \in \mathcal{L}} x_{A_i} c_{A_i} \leq C \end{aligned} \quad (9)$$

从 P0 可以看出, 约束条件 C1 表示分配给所有终端设备的计算资源总和小于或等于服务器的最大计算能力, C2 表示分配给所有终端设备的频谱资源小于或等于总频谱带宽, C3 表示任务卸载比例, C4 表示服务缓存决策, C5 表示边缘服务器可以缓存计算服务的最大容量。此外, P0 是由离散变量与连续变量组成的混合整数非线性规划问题, 很难求解, 因此需要设计一种新的优化方法对该问题进行求解。

3 服务缓存与资源分配联合优化策略

为了解决以上问题, 本文提出一种面向任务卸载与服务缓存问题的服务缓存与资源分配的联合优化策略, 简称为服务缓存与资源分配的联合优化策略。首先, 将原问题的连续变量与离散变量进行解耦, 变为 2 个子问题, 即服务缓存决策问题以及计算资源与通信资源联合优化问题; 其次, 在固定缓存决策基础上对计算与通信资源分配问题进行求解; 再次, 在得到优化的资源分配变量后对缓存决策进行优化; 最后, 通过循环迭代得到原问题的解。

3.1 计算资源与通信资源联合优化

首先, 初始化缓存决策; 随后, 对计算与通信资源进行联合优化。在给定缓存决策条件下, P0 变成由计算资源、带宽资源以及任务卸载比例等连续变量组成的函数。接下来将对连续变量进行求解。在给定初始缓存决策为 $x^{(0)}$ 后, 多终端时延优化函数表示为

$$\begin{aligned} y(\alpha, \beta, \theta, f) = & \sum_{i=1}^L x_{A_i}^{(0)} \left(\frac{\frac{c_{A_i}^{\text{input}} \beta_{A_i} s_i}{f_i^{\text{local}}} + \frac{c_{A_i}^{\text{input}} s_i (1 - \beta_{A_i})}{f_i^{\text{mec}}}}{\frac{c_{A_i}^{\text{input}} (1 - \beta_{A_i})}{\theta_{A_i} B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right)}} \right) + (1 - x_{A_i}^{(0)}) \cdot \\ & \left(\frac{\frac{c_{A_i}^{\text{input}} \alpha_{A_i} s_i}{f_i^{\text{local}}} + \frac{c_{A_i}^{\text{input}} (1 - \alpha_{A_i})}{R} + \frac{c_{A_i}^{\text{input}} (1 - \alpha_{A_i})}{\theta_{A_i} B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right)}}{R} \right) \end{aligned} \quad (10)$$

相应的优化问题表示为

$$\begin{aligned}
 \text{P1: } & \min_{\alpha, \beta, \theta, f} y(\alpha, \beta, \theta, f) \\
 \text{s.t. C6: } & \sum_{i \in \mathcal{L}} x_{A_i}^{(0)} f_i^{\text{mec}} \leq F \\
 \text{C2: } & \sum_{i \in \mathcal{L}} \theta_{A_i} \leq 1 \\
 \text{C3: } & 0 \leq \alpha_{A_i} \leq 1, 0 \leq \beta_{A_i} \leq 1, \forall i \in \mathcal{L} \quad (11)
 \end{aligned}$$

缓存决策确定后，其相应的约束条件同时发生变化，其约束条件将由 C1 变为 C6。为了避免除零情况的发生，对变量 θ_{A_i} 、 f_i^{mec} 加入常量 ε_1 、 ε_2 ，可表示为 $\bar{\theta}_{A_i} = (\theta_{A_i} + \varepsilon_1)^{-1}$ ， $\bar{f}_i^{\text{mec}} = (f_i^{\text{mec}} + \varepsilon_2)^{-1}$ 。最后将该变量代入式(10)以及对应的约束条件中，得到新的函数以及约束条件。相应的问题表示为

$$\begin{aligned}
 \text{P2: } & \min_{\alpha, \beta, \theta, f} \sum_{i=1}^L x_{A_i}^{(0)} \left(\frac{c_{A_i}^{\text{input}} \beta_{A_i} s_i}{f_i^{\text{local}}} + c_{A_i}^{\text{input}} s_i (1 - \beta_{A_i}) \bar{f}_i^{\text{mec}} + \frac{c_{A_i}^{\text{input}} (1 - \beta_{A_i}) \bar{\theta}_{A_i}}{B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right)} \right) + \\
 & (1 - x_{A_i}^{(0)}) \left(\frac{c_{A_i}^{\text{input}} \alpha_{A_i} s_i}{f_i^{\text{local}}} + \frac{c_{A_i}^{\text{input}} (1 - \alpha_{A_i}) \bar{\theta}_{A_i}}{B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right)} + \frac{c_{A_i}^{\text{input}} (1 - \alpha_{A_i})}{R} \right) \\
 \text{s.t. C7: } & \sum_{i \in \mathcal{L}} \frac{1}{\bar{f}_i^{\text{mec}}} - \varepsilon_2 \leq F, \quad k \in \mathcal{K} \\
 \text{C8: } & \sum_{i \in \mathcal{L}} \frac{1}{\bar{\theta}_{A_i}} - \varepsilon_1 \leq 1, \quad k \in \mathcal{K} \\
 \text{C3: } & 0 \leq \alpha_{A_i} \leq 1, 0 \leq \beta_{A_i} \leq 1, \forall i \in \mathcal{L} \quad (12)
 \end{aligned}$$

由 P2 可以看出，式(12)中存在 2 个变量相乘的乘积项，因此该问题是一个非凸优化问题。为了求解 P2，需要对以上问题进行再次变换。

3.2 基于 RLT 的问题转换

由于 P2 包含乘积项 $\alpha_{A_i} \bar{\theta}_{A_i}$ 、 $\beta_{A_i} \bar{f}_i^{\text{mec}}$ 、 $\beta_{A_i} \bar{\theta}_{A_i}$ ，因此需要通过 RLT 对上述问题进行变换。首先，引入 3 个辅助变量 μ_{A_i} 、 η_{A_i} 、 χ_{A_i} ，其中， $\mu_{A_i} = \alpha_{A_i} \bar{\theta}_{A_i}$ ， $\eta_{A_i} = \beta_{A_i} \bar{f}_i^{\text{mec}}$ ， $\chi_{A_i} = \beta_{A_i} \bar{\theta}_{A_i}$ 。将新的变量代入式(12)，P2 可以重写为

$$\begin{aligned}
 \text{P2: } & \min_{\alpha, \beta, \theta, f} \sum_{i \in \mathcal{L}} \left(x_{A_i}^{(0)} \frac{c_{A_i}^{\text{input}} \beta_{A_i} s_i}{f_i^{\text{local}}} + c_{A_i}^{\text{input}} s_i \bar{f}_i^{\text{mec}} - \right. \\
 & \left. s_i \eta_{A_i} \frac{c_{A_i}^{\text{input}} \bar{\theta}_{A_i} - c_{A_i}^{\text{input}} \chi_{A_i}}{B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right)} + (1 - x_{A_i}^{(0)}) \cdot \right.
 \end{aligned}$$

$$\left(\frac{c_{A_i}^{\text{input}} \alpha_{A_i} s_i}{f_i^{\text{local}}} + \frac{c_{A_i}^{\text{input}} \bar{\theta}_{A_i} - c_{A_i}^{\text{input}} \mu_{A_i}}{B \log \left(1 + \frac{|h_{i,B}|^2 P_{i,s} d^{-r}}{\sigma^2} \right)} + \frac{c_{A_i}^{\text{input}} (1 - \alpha_{A_i})}{R} \right) \quad (13)$$

对于新增变量 μ_{A_i} 、 η_{A_i} 、 χ_{A_i} ，以及 $0 \leq \alpha_{A_i} \leq 1$ 、 $0 \leq \beta_{A_i} \leq 1$ 、 $\frac{1}{F + \varepsilon_2} \leq \bar{f}_i^{\text{mec}} \leq \frac{1}{\varepsilon_2}$ 、 $\frac{1}{1 + \varepsilon_1} \leq \bar{\theta}_{A_i} \leq \frac{1}{\varepsilon_1}$ 等约束条件，可以得到 μ_{A_i} 的 RLT 因子积约束条件为

$$\begin{aligned}
 & \left\{ [\alpha_{A_i} - 0] \left[\bar{\theta}_{A_i} - \frac{1}{1 + \varepsilon_1} \right] \geq 0 \right\}_{\text{LS}} \\
 & \left\{ [\alpha_{A_i} - 0] \left[\frac{1}{\varepsilon_1} - \bar{\theta}_{A_i} \right] \geq 0 \right\}_{\text{LS}} \\
 & \left\{ [1 - \alpha_{A_i}] \left[\bar{\theta}_{A_i} - \frac{1}{1 + \varepsilon_1} \right] \geq 0 \right\}_{\text{LS}} \\
 & \left\{ [1 - \alpha_{A_i}] \left[\frac{1}{\varepsilon_1} - \bar{\theta}_{A_i} \right] \geq 0 \right\}_{\text{LS}} \quad (14)
 \end{aligned}$$

其中, $\{\cdot\}_{\text{LS}}$ 表示线性化步骤。将 $\mu_{A_i} = \alpha_{A_i} \bar{\theta}_{A_i}$ 代入式(14)可得

$$\begin{cases} \mu_{A_i} - \frac{\alpha_{A_i}}{1 + \varepsilon_1} \geq 0 \\ \frac{\alpha_{A_i}}{\varepsilon_1} - \mu_{A_i} \geq 0 \\ \bar{\theta}_{A_i} - \frac{1}{1 + \varepsilon_1} - \mu_{A_i} + \frac{\alpha_{A_i}}{1 + \varepsilon_1} \geq 0 \\ \frac{1}{\varepsilon_1} - \bar{\theta}_{A_i} - \alpha_{A_i} \frac{1}{\varepsilon_1} + \mu_{A_i} \geq 0 \end{cases} \quad (15)$$

其中, $\forall i \in \mathcal{L}$ 。同理, η_{A_i} 对应的 RLT 因子积约束条件为

$$\begin{cases} \left\{ [\beta_{A_i} - 0] \left[\bar{f}_i^{\text{mec}} - \frac{1}{F + \varepsilon_2} \right] \geq 0 \right\}_{\text{LS}} \\ \left\{ [\beta_{A_i} - 0] \left[\frac{1}{\varepsilon_2} - \bar{f}_i^{\text{mec}} \right] \geq 0 \right\}_{\text{LS}} \\ \left\{ [1 - \beta_{A_i}] \left[\bar{f}_i^{\text{mec}} - \frac{1}{F + \varepsilon_2} \right] \geq 0 \right\}_{\text{LS}} \\ \left\{ [1 - \beta_{A_i}] \left[\frac{1}{\varepsilon_2} - \bar{f}_i^{\text{mec}} \right] \geq 0 \right\}_{\text{LS}} \end{cases} \quad (16)$$

将 $\eta_{A_i} = \beta_{A_i} \bar{f}_i^{\text{mec}}$ 代入式(16)可得

$$\begin{cases} \eta_{A_i} - \frac{\beta_{A_i}}{F + \varepsilon_2} \geq 0 \\ \frac{\beta_{A_i}}{\varepsilon_2} - \eta_{A_i} \geq 0 \\ \bar{f}_i^{\text{mec}} - \frac{1}{F + \varepsilon_2} - \eta_{A_i} + \frac{\beta_{A_i}}{F + \varepsilon_2} \geq 0 \\ \frac{1}{\varepsilon_2} - \bar{f}_i^{\text{mec}} - \frac{\beta_{A_i}}{\varepsilon_2} + \eta_{A_i} \geq 0 \end{cases} \quad (17)$$

同理, χ_{A_i} 对应的 RLT 因子积约束条件为

$$\begin{cases} \left\{ [\beta_{A_i} - 0] \left[\bar{\theta}_{A_i} - \frac{1}{1 + \varepsilon_1} \right] \geq 0 \right\}_{\text{LS}} \\ \left\{ [\beta_{A_i} - 0] \left[\frac{1}{\varepsilon_1} - \bar{\theta}_{A_i} \right] \geq 0 \right\}_{\text{LS}} \\ \left\{ [1 - \beta_{A_i}] \left[\bar{\theta}_{A_i} - \frac{1}{1 + \varepsilon_1} \right] \geq 0 \right\}_{\text{LS}} \\ \left\{ [1 - \beta_{A_i}] \left[\frac{1}{\varepsilon_1} - \bar{\theta}_{A_i} \right] \geq 0 \right\}_{\text{LS}} \end{cases} \quad (18)$$

将 $\chi_{A_i} = \beta_{A_i} \bar{\theta}_{A_i}$ 代入式(18)可得

$$\begin{cases} \chi_{A_i} - \frac{\beta_{A_i}}{1 + \varepsilon_1} \geq 0 \\ \frac{\beta_{A_i}}{\varepsilon_1} - \chi_{A_i} \geq 0 \\ \bar{\theta}_{A_i} - \frac{1}{1 + \varepsilon_1} - \chi_{A_i} + \frac{\beta_{A_i}}{1 + \varepsilon_1} \geq 0 \\ \frac{1}{\varepsilon_1} - \bar{\theta}_{A_i} - \frac{\beta_{A_i}}{\varepsilon_1} + \chi_{A_i} \geq 0 \end{cases} \quad (19)$$

经过以上变换后, 得到新的优化问题 P3 为

$$\begin{aligned} \text{P3: } \min_{\alpha, \beta, \bar{\theta}, \bar{f}, \mu, \eta, \chi} y''(\alpha, \beta, \bar{\theta}, \bar{f}, \mu, \eta, \chi) \\ \text{s.t. C3, C7, C8} \\ \text{C9: 式(15)} \\ \text{C10: 式(17)} \\ \text{C11: 式(19)} \end{aligned} \quad (20)$$

P3 是一个凸优化问题, 因此可以通过拉格朗日乘子法以及梯度法对其进行求解。将得到的解 $\bar{\theta}$ 和 \bar{f} 代入 $\bar{\theta}_{A_i} = (\theta_{A_i} + \varepsilon_1)^{-1}$ 和 $\bar{f}_i^{\text{mec}} = (f_i^{\text{mec}} + \varepsilon_2)^{-1}$ 后, 可得原始变量的解。

3.3 缓存优化问题的解决

获得 P1 的解后, 下面解决缓存决策问题。首先, 将获得的资源分配方案 $\alpha^{(0)}$ 、 $\beta^{(0)}$ 、 $\theta^{(0)}$ 、 $f^{(0)}$ 代入 P0, 得到缓存决策问题 P4 为

$$\begin{aligned} \text{P4: } \min_x \sum_{i=1}^L x_{A_i} U_{A_i}^{\text{local_mec}} + (1 - x_{A_i}) U_{A_i}^{\text{local_cloud}} \\ \text{s.t. C4: } x_{A_i} \in \{0, 1\}, \quad \forall i \in \mathcal{L} \\ \text{C5: } \sum_{i \in \mathcal{L}} x_{A_i} c_{A_i} \leq C \end{aligned} \quad (21)$$

P4 是 0-1 问题, 可以通过分支定界法对以上问题进行求解。然而, 分支定界法在最坏的情况下需要运算 2^k 次, 当输入数据量较小时, 其复杂度是可接受的; 随着输入数据量的增大, 则很难在有效时间内获得近似最优解。为了降低时间复杂度, 本文将整型变量 x_{A_i} 松弛成 $0 \leq x_{A_i} \leq 1$ 的连续变量。此时, P4 变为凸优化问题, 可以通过内点法与 KKT (Karush-Kuhn-Tucker) 条件进行求解。随后采用四舍五入的方法对连续变量进行恢复, 这里将阈值简单设置为 0.5。

3.4 算法描述

基于上述分析可知, 边缘服务器通过阶段性获取当前接入终端的完整信息, 执行 JSCRA 优化策

略，并将缓存决策以及资源分配结果发送至终端设备，从而最小化所有设备任务的执行时延，具体步骤如算法 1 所示。

算法 1 服务缓存与资源分配联合优化算法

初始化 边缘服务的缓存策略 $x_{A_i}^{(0)}, \forall i \in \mathcal{L}$ ，终端发射功率 $p_{i,s}$ ，任务的输入数据大小 $c_{A_i}^{\text{input}}$ ，每比特输入数据所需的计算量 s_i ，最大迭代次数 t_{\max} ，算法终止条件 ξ 。

步骤 1 在已知服务缓存策略基础上，原问题变换为连续非线性问题，得到新问题 P2。

步骤 2 通过 RLT 技术对 P2 松弛，将非线性问题变为凸优化问题 P3。

步骤 3 采用内点法对 P3 进行求解，并得到在已知缓存决策 $x_{A_i}^{(0)}$ 条件下的资源分配方案 $\alpha, \beta, \bar{\theta}, \bar{f}, \mu, \eta, \chi$ 。

步骤 4 将步骤 3 得到的解代入 P4，通过松弛法以及凸优化方法得到相应的卸载决策 x_{A_i} 。

步骤 5 获取步骤 3 与步骤 4 中目标函数的差值，如果差值小于 ξ 或者迭代次数达到最大值 t_{\max} ，则停止计算；否则，对步骤 3 以及步骤 4 进行循环迭代。

4 仿真实验与性能评估

本节在 MATLAB 2017b 上进行仿真实验，并分析 JSCRA 优化策略的性能。此外，为了进行性能对比，本文对以下基准策略进行了仿真。

完全本地卸载 (LOC, local offloading completely) 策略：所有任务均在终端本地进行处理，此时假设本地终端存储任务相关的服务数据^[21]。

完全卸载至云 (COC, cloud offloading completely) 策略：任务相关服务数据全部存储在云服务器，此时任务完全在云服务器执行，而上行链路带宽资源均匀分配给每个终端设备^[18]。

随机缓存与资源分配 (RCRA, random caching and resource allocation) 策略：在存储约束条件下，边缘服务器采用随机缓存策略尽可能多地缓存服务内容并在其基础上对计算与带宽资源进行优化^[18]。

4.1 仿真参数设置

在仿真实验中，终端设备的数量为 10，随机分布在边长为 500 m 的正方形区域内。 $h_{i,B}$ 服从均值为 1 的指数分布^[22]，计算任务数据大小表示为 c_i 且服从 [100, 200]KB 的均匀分布， s_i 设置为 1 500 cycle/bit。仿真参数设置如表 1 所示。

表 1 仿真参数设置

参数	数值
终端设备数 L / 个	10
上行链路带宽 B / MHz	20
基站通信范围边长 d / m	500
计算任务输入数据 $c_{A_i}^{\text{input}}$ / KB	U[100,200]
计算服务种类 K / 种	10
移动终端发射功率 $P_{i,s}$ / dBm	23
路径损耗 r	4
噪声功率 δ^2 / (dBm·Hz ⁻¹)	-174
边缘服务器的计算能力 F / Megacycle	30 000
终端的计算能力 f_i^{local} / Megacycle	N(500,50)
执行一个任务所需的服务缓存 c_{A_i} / GB	U[30,80]
边缘服务器的最大存储能力 C / GB	500
服务器与远端云间传输速率 R / (Mbit·s ⁻¹)	1

4.2 算法复杂度与收敛性分析

由于原始问题 P0 是混合整数非线性规划问题，通过穷举法求解问题的复杂度为 $O(2^n)$ ，呈现指数级增长。所提算法将原始问题 P0 转换为 P3 与 P4，P3 为标准凸优化问题，可以通过内点法解决，内点法的复杂度为 $O(n^3)$ ；P4 为 0-1 问题，离散变量经过松弛后为凸优化问题，其算法复杂度为 $O(n^3)$ ，此外，交替迭代次数的复杂度为 $O(n)$ ，因此，所提算法总体复杂度为 $O(n^3)$ 。与穷举法相比，所提算法复杂度大大降低。

算法收敛过程如图 2 所示。本文将通过穷举法获得的值作为最优解。由于穷举法非常耗时，因此一般在小规模场景下使用。由图 2 可以看出，所提算法通过迭代 25 次快速收敛并接近最优解，这表明所提算法性能非常接近于穷举法的性能，并可以获得近似最优解。

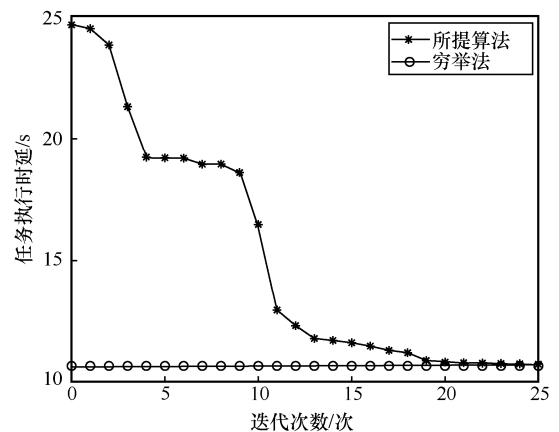


图 2 算法收敛过程

4.3 仿真结果和分析

本节仿真结果是在配置有 Intelcorei59400F 2.9 GHz 处理器和 16 GB RAM 的台式机上用 MATLAB 2017b 实现的。在 $N=10$, $C=300$ GB, $R=1$ 的条件下, 边缘服务器的计算能力和存储能力对任务执行时延的影响分别如图 3 和图 4 所示。

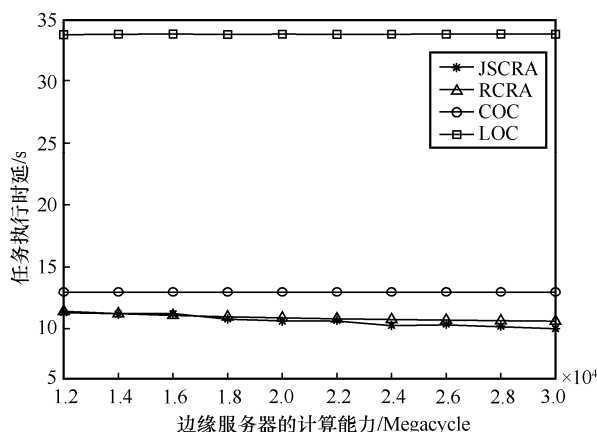


图 3 边缘服务器的计算能力对任务执行时延的影响

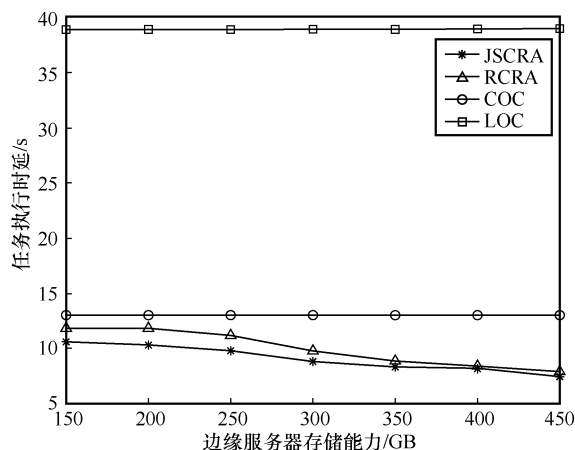


图 4 边缘服务器的存储能力对任务执行时延的影响

从图 3 可以看出, 随着边缘服务器的计算能力逐渐增大, 所提 JSCRA 优化策略与 RCRA 策略的任务执行时延逐渐减小, 而 LOC 策略以及 COC 策略任务执行时延保持不变。这是因为 LOC 策略的任务时延仅与本地计算能力有关, 任务执行时延与边缘服务器所拥有的资源没有关系, 所以 LOC 策略的所有任务执行时延保持不变。对于 COC 策略而言, 当任务在云服务器执行时, 其任务执行时延仅与无线带宽资源以及回程链路传输速率有关, 因此 COC 策略的任务执行时延不随边缘服务器计算能力的增大而改变。另外, 从图 3 还可以看出, JSCRA 优化策略在性能上略优于 RCRA 策略, 这是因为 JSCRA 优化策略对服务缓存进行合理优

化, 使更多任务可以在边缘服务器中执行, 因此性能优于 RCRA 策略。

从图 4 可以看出, 随着边缘服务器的存储能力逐渐提升, 所提策略与 RCRA 策略任务执行时延逐渐减小, 而 LOC 策略以及 COC 策略的任务执行时延保持不变。所提策略与 RCRA 策略相比可以减少 10% 的任务执行时延, 与 COC 策略相比可以减少 30% 的任务执行时延。LOC 策略的任务执行时延仅与本地计算能力有关, COC 策略的任务执行时延与带宽资源以及回程链路传输速率有关, 所以 LOC 策略与 COC 策略的任务执行时延不随边缘服务器存储能力的大小变化。此外, 从图 4 可以看出, 随着边缘服务器存储能力的增加, 与 COC 策略相比, 所提策略与 RCRA 策略性能增益增大。这是因为当边缘服务器存储能力较小时, 只能缓存部分计算服务, 剩余服务将存储在远端云服务器。此时, 大多数任务将在云服务中执行, 因此所提策略与 RCRA 策略的任务执行时延性能略优于 COC 策略。但是, 随着存储能力的增加, 边缘服务器可以缓存的服务内容增多, 远端执行任务比例逐渐降低, 此时边缘服务器计算资源被充分利用, 因此所提策略与 RCRA 策略的性能显著优于 COC 算法。

任务输入数据大小对任务执行时延的影响如图 5 所示。从图 5 可以看出, 任务输入数据越大, 所有策略的任务执行时延越大, 而所提策略的性能优于其他策略。这是因为随着输入数据量的增大, 任务所需的计算资源与通信资源将会增多, 所有策略的任务执行时延逐渐增大。另外, 由于所提策略对计算、通信以及存储资源进行联合优化, 因此其性能优于其他策略。

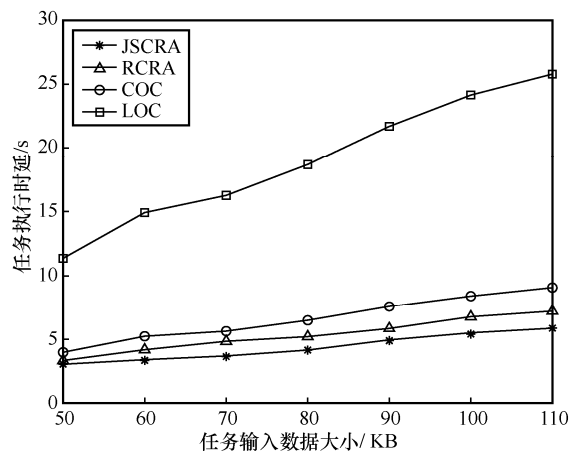


图 5 任务输入数据大小对任务执行时延的影响

边缘服务器与远端云间传输速率对任务执行时延的影响如图6所示。从图6可以看出,随着边缘服务器与远端云间的传输速率逐渐增大,所提策略与RCRA策略、COC策略的所有任务总时长逐渐减小,当 $R=5$ 时,上述3种策略性能基本相同。这是因为当边缘服务器与远端云间传输速率较小时,任务卸载至云服务器的时延大于在边缘服务器的执行时延,此时大部分任务将在边缘服务器执行,因此所提策略性能优于RCRA策略以及COC策略。随着传输速率增大,任务在云端执行的时延与任务在边缘服务器执行时延越来越接近,因此所提策略与RCRA策略以及COC策略的性能差异将越来越小,这也表明所提策略在传输速率较小时可以获得较高的性能增益。此外,LOC策略任务执行时延没有变化,这是因为任务本地时延与终端自身计算资源有关与边缘服务器,而与云端间的传输速率无关。

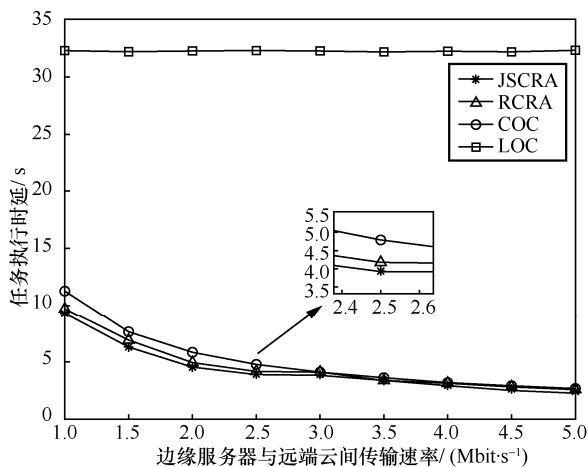


图6 边缘服务器与远端云间传输速率对任务执行时延的影响

5 结束语

在多终端、单服务节点以及云服务器组成的集中式移动边缘网络下,针对任务卸载与服务缓存耦合性对任务时延的影响,本文提出了一种服务缓存与资源分配的联合优化策略。首先,在计算、通信以及存储容量约束下,建立服务缓存与资源分配联合优化问题。随后,将原问题进行分解为2个子问题并通过重构线性化方法、松弛以及凸优化方法对其进行交替迭代优化。仿真结果表明,与RCRA策略、LOC策略和COC策略相比,所提策略可以有效地降低系统时延并具有良好的收敛性。

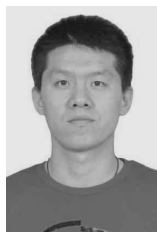
参考文献:

- [1] LIU F M, SHU P, JIN H, et al. Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications[J]. IEEE Wireless Communications, 2013, 20(3): 14-22.
- [2] ZHOU Y Q, TIAN L, LIU L, et al. Fog computing enabled future mobile communication networks: a convergence of communication and computing[J]. IEEE Communications Magazine, 2019, 57(5): 20-27.
- [3] DINH H T, LEE C, NIYATO D, et al. A survey of mobile cloud computing: architecture, applications, and approaches[J]. Wireless Communications and Mobile Computing, 2013, 13(18): 1587-1611.
- [4] XU Y J, GUI G, GACANIN H, et al. A survey on resource allocation for 5G heterogeneous networks: current research, future trends, and challenges[J]. of IEEE Communications Surveys & Tutorials, 2021, 23(2): 668-695.
- [5] ZHANG W W, WEN Y G, WU D O. Energy-efficient scheduling policy for collaborative execution in mobile cloud computing[C]//Proceedings of IEEE INFOCOM. Piscataway: IEEE Press, 2013: 190-194.
- [6] ZHANG W W, WEN Y G, WU D O. Collaborative task execution in mobile cloud computing under a stochastic wireless channel[J]. IEEE Transactions on Wireless Communications, 2015, 14(1): 81-93.
- [7] XU Y J, XIE H, LI D, et al. Energy-efficient beamforming for heterogeneous industrial IoT networks with phase and distortion noises[J]. IEEE Transactions on Industrial Informatics, 2022, 18(11): 7423-7434.
- [8] XU Y J, XIE H, WU Q Q, et al. Robust max-min energy efficiency for RIS-aided HetNets with distortion noises[J]. IEEE Transactions on Communications, 2022, 70(2): 1457-1471.
- [9] XU Y J, GU B W, LI D, et al. Resource allocation for secure SWIPT-enabled D2D communications with α fairness[J]. IEEE Transactions on Vehicular Technology, 2022, 71(1): 1101-1106.
- [10] GUO S T, XIAO B, YANG Y Y, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C]//Proceedings of the 35th Annual IEEE International Conference on Computer Communications. Piscataway: IEEE Press, 2016: 1-9.
- [11] LONG L, LIU Z C, ZHOU Y Q, et al. Delay optimized computation offloading and resource allocation for mobile edge computing[C]//Proceedings of IEEE 90th Vehicular Technology Conference. Piscataway: IEEE Press, 2019: 1-5.
- [12] ZHAO H Y, WANG Y, SUN R J. Task proactive caching based computation offloading and resource allocation in mobile-edge computing systems[C]//Proceedings of 14th International Wireless Communications & Mobile Computing Conference (IWCMC). Piscataway: IEEE Press, 2018: 232-237.
- [13] LIU M T, YU F R, TENG Y L, et al. Joint computation offloading and content caching for wireless blockchain networks[C]//Proceedings of IEEE Conference on Computer Communications Workshops. Piscataway: IEEE Press, 2018: 517-522.
- [14] XU Z C, ZHOU L Z, CHI-KIN C S, et al. Collaborate or separate? distributed service caching in mobile edge clouds[C]//Proceedings of IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 2066-2075.
- [15] LIANG Y, GE J D, ZHANG S, et al. A utility-based optimization

framework for edge service entity caching[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(11): 2384-2395.

- [16] CHEN J Y, WU H Q, YANG P, et al. Cooperative edge caching with location-based and popular contents for vehicular networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(9): 10291-10305.
- [17] LIU Y Z, WANG S H, OBAIDAT M S, et al. Service chain caching and workload scheduling in mobile edge computing[J]. IEEE Systems Journal, 2022, 16(3): 4389-4400.
- [18] TRAN T X, CHAN K, POMPILI D. COSTA: cost-aware service caching and task offloading assignment in mobile-edge computing[C]//Proceedings of 16th Annual IEEE International Conference on Sensing, Communication, and Networking. Piscataway: IEEE Press, 2019: 1-9.
- [19] CAO X W, WANG F, XU J, et al. Joint computation and communication cooperation for energy-efficient mobile edge computing[J]. IEEE Internet of Things Journal, 2019, 6(3): 4188-4200.
- [20] WANG F, XU J, WANG X, et al. Joint offloading and computing optimization in wireless powered mobile-edge computing systems[C]//Proceedings of 2017 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2017: 1-6.
- [21] ZHANG Y, NIYATO D, WANG P. Offloading in mobile cloudlet systems with intermittent connectivity[J]. IEEE Transactions on Mobile Computing, 2015, 14(12): 2516-2529.
- [22] ZHANG J, XIA W W, ZHANG Y Y, et al. Joint offloading and resource allocation optimization for mobile edge computing[C]//Proceedings of 2017 IEEE Global Communications Conference. Piscataway: IEEE Press, 2017: 1-6.

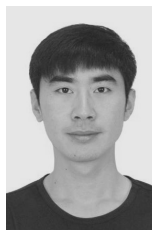
[作者简介]



龙隆(1988-), 男, 山西长治人, 博士, 中国科学院计算技术研究所工程师, 主要研究方向为移动通信、移动边缘计算、计算与通信融合、农业模型与算法等。



刘子辰(1984-), 男, 山东临沂人, 博士, 中国科学院计算技术研究所工程师, 主要研究方向为农机大数据、农业生产执行系统、图数据库系统。



陆在旺(1992-), 男, 山东潍坊人, 中国科学院大学博士生, 主要研究方向为智能通信、强化学习、多机调度、农业机器人等。



张玉成(1981-), 男, 江苏淮安人, 博士, 中国科学院计算技术研究所正高级工程师, 主要研究方向为智能通信、智能农机、复杂农业系统控制理论与方法。



李蕾(1987-), 女, 河南洛阳人, 中国科学院大学博士生, 主要研究方向为人工智能、智慧农业等。