

基于时延和能耗约束的感知数据协作卸载策略研究

袁培燕^{1,2}, 邵赛珂^{1,2}, 魏然^{1,2}, 张俊娜^{1,2}, 赵晓焱^{1,2}

(1. 河南师范大学计算机与信息工程学院, 河南 新乡 453007;

2. 河南师范大学教学资源与教育质量评估大数据河南省工程实验室, 河南 新乡 453007)

摘要: 研究了物联网感知数据边缘卸载问题, 即多个边缘节点相互协作, 将原本需要发送给云中心的感知数据全部或部分卸载, 以保护数据隐私与提升用户体验。在协作卸载过程中, 感知数据传输以及边缘节点之间的信息交互会消耗系统资源, 产生协作代价。如何在保持较低协作代价的基础上提高感知数据的卸载比例是一个具有挑战性的问题。首先, 将该问题表述为一个满足网络时延和系统能耗约束的感知数据卸载比例和协作规模联合优化问题。其次, 提出了一种基于约束投影和变量分裂的分布式交替方向乘子法 (ADMM, alternating direction method of multipliers) 进行求解。最后, 使用 MATLAB 进行仿真实验, 数值结果表明, 与分布式优化算法 (DOA, distributed optimization algorithm)、公平合作算法 (FCA, fairness cooperation algorithm) 和多子任务到多服务器卸载方案 (MTMS, multi-subtasks-to-multi-servers offloading scheme) 相比, 所提方法在网络时延和能耗上均有较大优化。

关键词: 协同边缘计算; 数据卸载; 系统能耗; 网络时延; 分布式 ADMM

中图分类号: TP391

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2023.00324

Research on the cooperative offloading strategy of sensory data based on delay and energy constraints

YUAN Peiyan^{1,2}, SHAO Saikē^{1,2}, WEI Ran^{1,2}, ZHANG Junna^{1,2}, ZHAO Xiaoyan^{1,2}

1. College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

2. Big Data Engineering Laboratory for Teaching Resources & Assessment of Education Quality, Henan Normal University, Xinxiang 453007, China

Abstract: The edge offloading of the internet of things (IoT) sensing data was investigated. Multiple edge servers cooperatively offload all or part of the sensing data initially sent to the cloud center, which protects data privacy and improves user experience. In the process of cooperative offloading, the transmission of the sensing data and the information exchange among edge servers will consume system resources, resulting in the cost of cooperation. How to maximize the offloading ratio of the sensing data while maintaining a low collaboration cost is a challenging problem. A joint optimization problem of sensing data offload ratio and cooperative scale satisfying the constraints of network delay and system energy consumption was formulated. Subsequently, a distributed alternating direction method of multipliers (ADMM) via constraint projection and variable splitting was proposed to solve the problem. Finally, simulation experiments were carried out on MATLAB. Numerical results show that the proposed method improved the network delay and energy consumption compared to the fairness cooperation algorithm (FCA), the distributed optimization algorithm (DOA), and multi-subtasks-to-multi-servers offloading scheme (MTMS) algorithm.

Key words: collaborative edge computing, data offloading, system energy consumption, network delay, distributed ADMM

收稿日期: 2022-09-07; 修回日期: 2023-01-03

通信作者: 袁培燕, peiyan@htu.cn

基金项目: 国家自然科学基金资助项目 (No.62072159, No.U1804164, No.61902112); 河南省教育厅重点项目 (No.19A510015, No.20A520019, No.20A520020)

Foundation Items: The National Natural Science Foundation of China (No.62072159, No.U1804164 No.61902112), The Key Project of the Education Department of Henan Province (No.19A510015, No.20A520019, No.20A520020)

0 引言

随着智能物联网移动设备（如智能手机、可穿戴设备）的快速普及，大量感知数据对骨干网络的流量分配带来了严重挑战。传统的云计算模式已经很难满足这种计算密集型应用和服务的需求^[1]。移动边缘计算作为一种新兴的计算范式，通过在靠近数据源的网络边缘部署服务器，将计算、存储等功能从云中心下沉到网络边缘，有助于把计算密集和时延敏感型任务卸载到边缘服务器进行处理，从而满足用户对任务实时响应和动态处理的需求^[2-6]。

考虑单个边缘设备受限于自身资源，卸载能力不足，多个设备的协作卸载可以有效提升计算性能。这种边缘协同计算既包括终端设备、边缘服务器和云中心之间相互协作的垂直协作^[7-10]，也包括边缘计算层中边缘服务器之间的水平协作^[11-13]。协同边缘计算既解决了垂直协作忽略边缘服务器之间的性能差异，导致边缘服务器负载不均衡的问题，也解决了水平协作忽略云中心强大的算力资源，导致处理时延较大的问题。目前已有很多这方面的研究^[14-19]。文献[15]以最小化所有任务的平均完成时间为目标建立协同边缘计算的联合多跳、多任务部分计算卸载和网络流调度问题，并采用一种联合部分卸载和流量调度的启发式算法，通过考虑设备的等待时间和网络流的启动时间决定部分卸载比例。文献[16]将边缘计算网络划分为 3 层，即迁移协作层、计算共享层和远程辅助层。通过水平和垂直协作解决边缘网络中的突发负载问题。

更重要的是，协同边缘计算虽然可以提高边缘网络计算卸载的能力，但是边缘设备受限于自身的硬件能力，无法保证任务全部卸载到边缘端，故任务的部分卸载也成为了关注重点^[20-22]。并且考虑在协同边缘计算中、随着协作规模的扩大，传输时延和处理能耗也会随之增加，这将影响网络性能。文献[23]通过约束边缘设备的协作规模来减少额外的网络损耗，但协作规模的限制会影响边缘设备对任务的卸载比例。本文综合考虑感知数据的卸载比例和边缘节点的协作规模，构建了一个协同边缘计算模型，每个边缘服务器将其部分或者全部工作负载转发给其他边缘服务器，提高边缘网络的计算卸载能力。同时，本文提出了一个满足网络时延和系统能耗约束的感知数据卸载比例和协作规模联合优

化问题，采用分布式交替方向乘子法（ADMM, alternating direction method of multipliers）进行求解，避免边缘服务器私有信息的共享导致隐私泄露问题。本文的主要贡献包括以下 3 个方面。

1) 本文构建了协同边缘计算模型，边缘节点通过协作卸载以共享资源。模型分别考虑了移动设备、边缘服务器和云中心之间的垂直协作以及边缘服务器之间的水平协作，在提高边缘网络计算卸载能力的同时，避免边缘服务器之间性能差异所带来的负载不均衡问题以及对云中心强大的算力资源的忽视。

2) 为解决多边缘服务器之间相互协作导致的高时延和高功耗问题，本文综合考虑感知数据的卸载比例和边缘节点的协作比例，以在网络时延和系统能耗约束下最大化卸载比例和协作比例之和。

3) 本文提出了一种基于约束投影和变量分裂的分布式交替方向乘子法（分布式 ADMM）对感知数据的卸载比例和边缘节点的协作比例进行优化，其分布式框架对变量的交替求解，不需要边缘节点之间共享参数信息，可以避免边缘节点私有信息被泄露。

1 相关工作

现有工作对比见表 1，目前大部分工作关于计算卸载的研究着重于通过节约系统成本（如能耗、时延等）提高用户服务质量。例如，文献[24-26]的研究工作在最小化时延方面取得了显著的效果，但仅考虑了任务的完全卸载。文献[24]考虑资源受限的物联网感知设备的低时延处理需求，提出了最小时延卸载决策和资源分配方案，以在能耗约束下最小化任务的完成时间。文献[25]研究了一种多用户协同边缘计算系统，在该系统中，本地移动用户可以将多个独立的计算任务卸载到多个附近的边缘设备，通过优化任务的分配减少计算时延。从降低系统能耗的角度出发，文献[27]提出了一种协同计算框架，该框架考虑了雾节点和云中心之间的垂直和水平协作，以在服务时延约束下最小化所有移动设备的总能耗。考虑双目标的优化，文献[28-29]以优化时延和能耗加权和为目标。文献[28]引入基于层次分析法的服务质量评估框架，对具有不同服务质量需求的各类物联网设备进行分析，以最大限度地减少雾计算网络的任务进程时延和能耗开销。文献[29]研究了移动边缘计算系统中移动设备执行时延与能耗之间的权衡问题，利用拉格朗日对

表 1 现有工作对比

	部分卸载	协作规模	以时延为目标	以能耗为目标	以时延和能耗加权和为目标
文献[22]	是	否	是	否	否
文献[20]	是	否	是	否	否
文献[25]	否	否	是	否	否
文献[29]	否	否	否	否	是
文献[24]	否	否	是	否	否
文献[28]	否	否	否	否	是
文献[27]	否	否	否	是	否
文献[23]	否	是	否	否	否

偶分解,提出了一种最小化执行时延和能耗加权的分布式算法。

文献[20-22]从最小化时延方面开展相关工作,同时兼顾任务的部分卸载,文献[20]提出了一个基于部分卸载策略的优化问题,通过优化卸载比例和资源分配,制定部分卸载策略,使多用户移动边缘计算(MEC, mobile edge computing)系统中所有用户的加权和时延最小。文献[22]研究了雾计算网络的工作负载卸载问题,一组雾节点可以将原本针对云数据中心的部分或全部工作负载卸载,以减少用户的响应时延。并在此基础上考虑协同雾计算网络,多个雾节点可以相互帮助,共同分担云数据中心的工作负载,以进一步提高用户的体验质量(QOE, quality of experience)。但是参与协作的边缘服务器的增加,会导致

额外的网络开销,文献[23]着重于通过约束边缘服务器的协作规模来减少通信成本,但协作规模的约束,会影响计算卸载的能力。

不同的文献分别考虑了任务的全部卸载、部分卸载或边缘节点的协作规模,据笔者所知,目前还没有文献综合考虑感知数据的卸载比例与协作规模,并且相对于大部分文献以时延、能耗或时延与能耗的加权和为目标函数进行问题的求解,本文是以最大化卸载比例与协作比例的和作为目标函数,并采用分布式ADMM进行求解,在时延和能耗的约束下求解感知数据的最佳卸载比例和边缘服务器的最优协作规模。

2 系统模型架构

本文考虑的移动边缘计算架构如图 1 所示,包

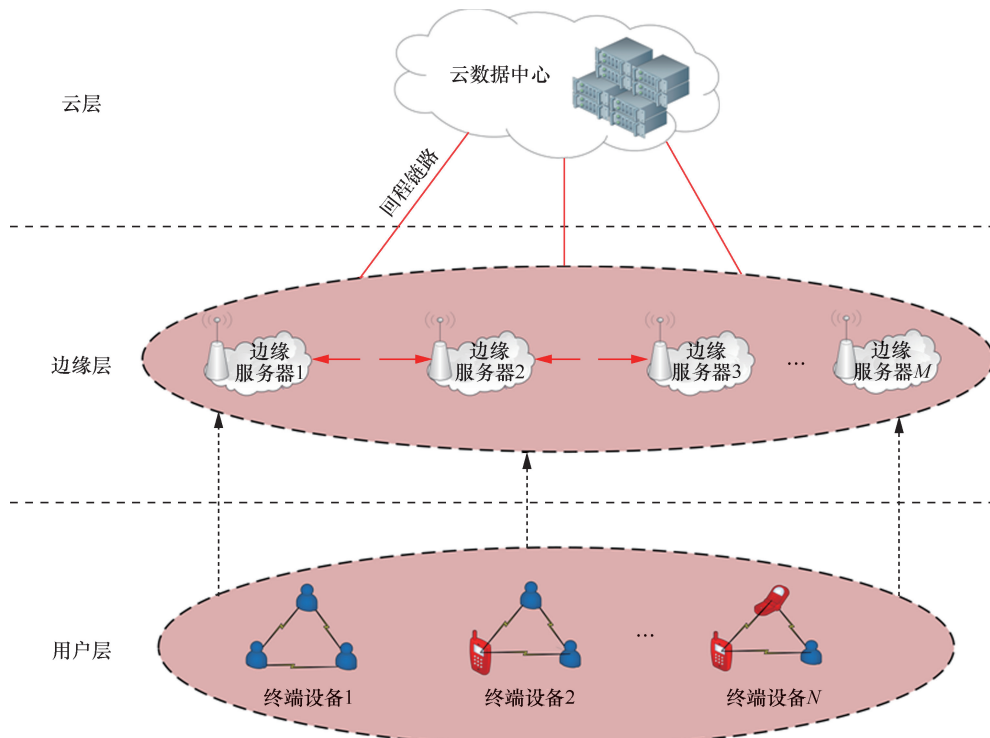


图 1 移动边缘计算架构

括3层网络结构,即用户层、边缘层和云层,用户层由 N 个物联网感知终端组成,边缘层包含 M 个边缘服务器,云层包括大规模的云数据中心。在该系统中,用户层的终端设备是具有感知功能的物联网设备,边缘层的边缘服务器具有一定的计算和存储功能(如蜂窝网络中的蜂窝基站)。需要注意的是,物联网感知终端通过无线通信信道连接边缘服务器,边缘服务器通过回程链路连接云中心。

本文将用户层的终端设备集定义为 $\omega=[1,2,\dots,N]$,边缘层中边缘服务器集定义为 $\xi=[1,2,\dots,M]$ 。本文定义 α_i^k 为边缘服务器 ξ_i 卸载第 k 类感知数据的比例, $\alpha_i^k=1$ 表示边缘服务器 ξ_i 完全卸载第 k 类感知数据, $\alpha_i^k=0$ 表示边缘服务器 ξ_i 将第 k 类感知数据全部转发给云中心处理。考虑边缘服务器之间的协作卸载,进一步地,令 α_{ii}^k 表示边缘服务器 ξ_i 本身对感知数据的卸载比例, α_{ij}^k 表示边缘服务器 ξ_j 卸载来自边缘服务器 ξ_i 的第 k 类感知数据的卸载比例。则有 $\alpha_i^k = \alpha_{ii}^k + \sum_{j \in A_k} \alpha_{ij}^k$,这里 A_k 表示配置有第 k 类应用的边缘服务器集合,卸载比例 α_i^k 表示边缘服务器 ξ_i 和其他部署有第 k 类应用的边缘服务器协同卸载第 k 类感知数据的卸载比例。

本文将全体协作节点构成的网络空间称为协作域,故定义 ϕ_i^k 为边缘服务器 ξ_i 针对第 k 类感知数据、选择协作的边缘服务器数量占全部边缘服务器数量的比值。需要注意的是,协作域内的边缘服务器包括配置有第 k 类应用的边缘服务器以及在数据传输过程中参与转发的边缘服务器。

3 问题建模与求解

3.1 问题的建模

3.1.1 网络时延

网络时延包括感知数据传输时延和在边缘服务器上的处理时延。本文令 τ_{ki} 、 τ_{ij} 、 τ_{ic} 分别表示终端设备和边缘服务器、边缘服务器之间、边缘服务器和云数据中心的感知数据传输时延, \bar{f}_i 、 \bar{f}_c 分别表示边缘服务器和云数据中心处理单位感知数据的时钟周期, f_k 、 f_c^k 分别表示边缘服务器和云数据中心为第 k 类应用分配的CPU资源, d_k 表示第 k 类单位感知数据的大小。考虑边缘服务器受限于自身的

硬件能力,无法将接收到的所有感知数据进行处理,故边缘层的边缘服务器协同处理 α_i^k 部分的感知数据,即当 $0 < \alpha_i^k \leq 1$, $\alpha_i^k = \alpha_{ii}^k + \sum_{j \in A_k} \alpha_{ij}^k$,边缘服务器 ξ_i 和其他部署有第 k 类应用的边缘服务器 ξ_j 协同处理 α_i^k 部分的感知数据,剩余 $(1-\alpha_i^k)$ 部分的感知数据由云中心处理,相应的网络时延可以写为

$$D_i^k = \tau_{ki} + \alpha_{ii}^k \frac{\bar{f}_i \mu_i^k d_k}{f_k} + \sum_{j \in A_k} \alpha_{ij}^k \left(\tau_{ij} + \frac{\bar{f}_j \mu_j^k d_k}{f_k} \right) + (1 - \alpha_i^k) \left(\tau_{ic} + \frac{\bar{f}_c \mu_c^k d_k}{f_c^k} \right) \quad (1)$$

3.1.2 系统能耗

这里的系统能耗主要关注边缘服务器的能量消耗。边缘服务器的能量消耗主要包括两部分:卸载节点处理工作负载的能耗与中间节点转发和接收感知数据的能耗。本文令 E_s 、 E_r 分别表示边缘服务器转发和接收单位感知数据的能耗,令 p_i 表示边缘服务器 ξ_i 的CPU的运转功率,令 E_i 、 E_i 分别表示边缘服务器处理工作负载的能耗与转发和接收的能耗。

边缘服务器处理工作负载的能耗为

$$E_i = p_i \times \alpha_{ii}^k \frac{\bar{f}_i \mu_i^k d_k}{f_k} \quad (2)$$

边缘服务器转发和接收能耗上限为

$$E_i = (E_s + E_r) \mu_i^k d_k \quad (3)$$

因为协作域内的边缘服务器采用生成树算法对其进行组织,所以由生成树的结构可知, $M \times \phi_i^k$ 个边缘服务器一共生成了 $(M \times \phi_i^k - 1)$ 条边,其中 M 为边缘服务器的总数量,则中间节点的转发和接收总能耗上限为

$$2(M \times \phi_i^k - 1)E_i \quad (4)$$

结合式(2)~式(4)可以得出,系统总能耗为

$$E_i^k = \sum_{i \in A_k} E_i + 2(M \times \phi_i^k - 1)E_i \quad (5)$$

本文的目标是在网络时延和系统能耗约束下,提高感知数据的卸载比例 α_i^k ,同时确定最优的协作规模 ϕ_i^k ,保持较小的协作代价并提高成功查找协作边缘服务器的概率。本文以最大化感知数据卸载比例与边缘服务器协作比例的和为目标函数,联合优化问题为

$$\begin{aligned}
& \max_{\alpha, \phi} \sum_{k \leq K} \sum_{i \leq M} (\alpha_i^k + \phi_i^k) \\
& \text{s.t.} \quad \sum_{k \leq K} \sum_{i \leq M} D_i^k \leq D \\
& \quad \sum_{k \leq K} \sum_{i \leq M} E_i^k \leq E \\
& \quad \sum_{k \leq K} \mu_i^k \leq \mu_i, 1 \leq i \leq M
\end{aligned} \quad (6)$$

其中，前两个约束条件分别表示网络时延与系统能耗不能超过设定的阈值， D_i^k 表示边缘服务器 ξ_i 处理第 k 类感知数据的网络时延， E_i^k 表示边缘服务器 ξ_i 处理第 k 类感知数据的能耗。需要注意的是，虽然不同边缘服务器可以进行并行处理，但是当边缘服务器 ξ_i 查找可以协作的其他边缘服务器时，其采用广度优先搜索算法，查找过程存在先后顺序，即不可能同时搜索到所有的协作边缘服务器，因此协作边缘服务器处理感知数据时，在时间上存在先后次序，所以目标函数第一个约束是把所有时延加起来，这与边缘服务器可以进行并行处理不存在冲突。第 3 个约束条件表示边缘服务器处理各类感知数据总量不能超过其服务速率上限，其中 μ_i^k 为边缘服务器 ξ_i 收到的第 k 类感知数据流。

3.2 问题的求解

在移动边缘计算中，边缘服务器之间的协作卸载虽然有助于提高感知数据的卸载比例，但是在协作过程中，感知数据传输以及节点之间信息交互均会产生额外的传输时延和处理能耗，因此让每个边缘服务器确定协作规模的最优大小和感知数据的最佳卸载比例，是实现良好的网络性能的前提，故期望设计一种轻量级和分布式的算法。本文采用 ADMM 进行问题的求解，但考虑 ADMM 是一种集中式优化策略，每个边缘服务器需要与其他边缘服务器交换参数信息来计算相关变量，这将导致隐私泄露的问题，并加重系统资源消耗。故提出分布式 ADMM，将式(6)分解为 N 个子问题，每个子问题可由边缘服务器利用其私有的信息进行求解，避免因其私有信息的共享出现隐私泄露的问题。但是考虑 ADMM 的局限性，将分布式 ADMM 应用到问题的求解中，需要考虑以下几个问题。

1) 传统的 ADMM 只能用于求解具有等式约束的优化问题。但是，目标函数的优化问题包含不等式约束，因此不能直接用 ADMM 求解。

2) 传统的 ADMM 是一种集中式优化策略，它要求每个节点的私有信息与其他节点共享，但节点信息的共享可能导致用户隐私泄露的风险，因此不能利用传统的方法优化网络性能，故本文提出了分布式 ADMM，其需要将 ADMM 从集中式转换为分布式方案，避免边缘节点之间信息的共享，使每个边缘节点利用其私有信息便可独立计算。

为了解决问题 1)，添加一个松弛变量 β_b ($b=1,2$)，并将其整合到目标函数中。松弛变量 β_b 的求解如下。

首先，针对式(6)的约束条件可以改写为

$$\begin{aligned}
g_1(\alpha_i^k, \phi_i^k) &= D - \\
& \sum_{k \leq K} \sum_{i \leq M} \left(\tau_{ki} + \alpha_{ii}^k \frac{\bar{f}_i \mu_i^k d_k}{f_k} + \sum_{j \in A_k} \alpha_{ij}^k \left(\tau_{ij} + \frac{\bar{f}_j \mu_i^k d_k}{f_k} \right) + \right. \\
& \left. (1 - \alpha_i^k) \left(\tau_{ic} + \frac{\bar{f}_c \mu_i^k d_k}{f_c} \right) \right) \\
g_2(\alpha_i^k, \phi_i^k) &= E - \sum_{k \leq K} \sum_{i \leq M} \left(\sum_{j \in A_k} \left(p_j \times \alpha_{ij}^k \frac{\bar{f}_j \mu_i^k d_k}{f_k} \right) + \right. \\
& \left. 2(M \times \phi_i^k - 1)(E_s + E_r) \mu_i^k d_k \right)
\end{aligned} \quad (7)$$

然后，引入松弛变量 β_b 将不等式约束问题转化为等式约束问题。

$$\begin{aligned}
& \min f(\alpha_i^k, \phi_i^k) = - \sum_{k \leq K} \sum_{i \leq M} (\alpha_i^k + \phi_i^k) \\
& \text{s.t.} \\
& g_1(\alpha_i^k, \phi_i^k) - \beta_1 = 0, \forall i = 1, 2, \dots, M \text{ 且 } \forall k = 1, 2, \dots, K \\
& g_2(\alpha_i^k, \phi_i^k) - \beta_2 = 0, \forall i = 1, 2, \dots, M \text{ 且 } \forall k = 1, 2, \dots, K
\end{aligned} \quad (8)$$

其次，将式(8)转化为不含约束条件的增广拉格朗日函数。

$$\begin{aligned}
L(\alpha_i^k, \phi_i^k, \lambda) &= f(\alpha_i^k, \phi_i^k) + \\
& \sum_{b=1}^2 \lambda \left[g_b(\alpha_i^k, \phi_i^k) - \beta_b \right] + \frac{\rho}{2} \sum_{b=1}^2 \left[g_b(\alpha_i^k, \phi_i^k) - \beta_b \right]^2 = \\
& f(\alpha_i^k, \phi_i^k) + \sum_{b=1}^2 \lambda \left[\beta_b - g_b(\alpha_i^k, \phi_i^k) \right] + \\
& \frac{\rho}{2} \sum_{b=1}^2 \left[\beta_b - g_b(\alpha_i^k, \phi_i^k) \right]^2 = f(\alpha_i^k, \phi_i^k) + \\
& \sum_{b=1}^2 \left\{ \frac{\rho}{2} \left[\beta_b - g_b(\alpha_i^k, \phi_i^k) \right]^2 + \lambda \left[\beta_b - g_b(\alpha_i^k, \phi_i^k) \right] \right\} =
\end{aligned}$$

$$f(\alpha_i^k, \phi_i^k) + \sum_{b=1}^2 \frac{\rho}{2} \left\{ \left[\beta_b - g_b(\alpha_i^k, \phi_i^k) + \frac{\lambda}{\rho} \right]^2 - \left(\frac{\lambda}{\rho} \right)^2 \right\} =$$

$$f(\alpha_i^k, \phi_i^k) + \sum_{b=1}^2 \frac{\rho}{2} \left\{ \left[\beta_b - \left[g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right] \right]^2 - \left(\frac{\lambda}{\rho} \right)^2 \right\} \quad (9)$$

若变量 α_i^k 、 ϕ_i^k 的值确定, $L(\alpha_i^k, \phi_i^k, \lambda)$ 要取得极值, 必有 $\beta_b - \left[g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right]$ 的值取得最小, 故将式(9)的结果进行如下分析。

当 $g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \geq 0$, 有 $\beta_b = g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho}$;

当 $g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} < 0$, 有 $\beta_b = 0$ 。

根据上述分析可以确定松弛变量 β_b 的表达式为

$$\beta_b = \max \left\{ 0, g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right\} =$$

$$\frac{1}{\rho} \max \left\{ 0, \rho \times \left[g_b(\alpha_i^k, \phi_i^k) \right] - \lambda \right\} \quad (10)$$

最后, 将松弛变量 β_b 的表达式 (即式(10)) 推导的结果代入目标式转化后的增广拉格朗日函数 (即式(9)), 得出不含松弛变量的增广拉格朗日函数为

$$L(\alpha_i^k, \phi_i^k, \lambda) = f(\alpha_i^k, \phi_i^k) + \sum_{b=1}^2 \lambda \left[g_b(\alpha_i^k, \phi_i^k) - \beta_b \right] +$$

$$\frac{\rho}{2} \sum_{b=1}^2 \left[g_b(\alpha_i^k, \phi_i^k) - \beta_b \right]^2 = f(\alpha_i^k, \phi_i^k) +$$

$$\sum_{b=1}^2 \frac{\rho}{2} \left\{ \left[\beta_b - \left[g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right] \right]^2 - \left(\frac{\lambda}{\rho} \right)^2 \right\} =$$

$$f(\alpha_i^k, \phi_i^k) + \sum_{b=1}^2 \frac{\rho}{2} \left\{ \left[\max \left\{ 0, g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right\} - \right. \right.$$

$$\left. \left[g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right] \right]^2 - \left(\frac{\lambda}{\rho} \right)^2 \right\} = f(\alpha_i^k, \phi_i^k) +$$

$$\sum_{b=1}^2 \frac{\rho}{2} \left\{ \left[\max \left\{ 0, - \left[g_b(\alpha_i^k, \phi_i^k) - \frac{\lambda}{\rho} \right] \right\} \right]^2 - \left(\frac{\lambda}{\rho} \right)^2 \right\} =$$

$$f(\alpha_i^k, \phi_i^k) + \frac{1}{2\rho} \sum_{b=1}^2 \left[\left[\max \left\{ 0, \lambda - \rho \times g_b(\alpha_i^k, \phi_i^k) \right\} \right]^2 - \lambda^2 \right] \quad (11)$$

为了求解问题 2), 需要证明式(6)增广拉格朗日函数是可分的。式(11)是式(6)不含松弛变量的增广

拉格朗日函数, 因为式(11)的值仅和边缘服务器 ξ_i 以及感知数据 k 有关, 与其他边缘服务器相关的变量相互独立。这证明优化问题的增广拉格朗日函数可以分解为 N 个子问题, 每个子问题可以由边缘服务器利用其私人参数信息进行求解。

分布式 ADMM 见算法 1, 每个边缘服务器 ξ_i 将私有参数 α_i 发送到云中心, 云中心通过迭代更新生成变量 ϕ_i 和 λ_i , 并将其发送给边缘服务器 ξ_i 。通过这种方式, 每个边缘服务器不需要知道其他节点的私有信息, 便可以求得最优解, 并避免隐私的泄露。

算法 1 分布式 ADMM

初始化 α_i^0, ϕ_i^0 and λ_i^0

for $k = 0, 1, \dots$

Edge node:

$$\alpha_i^{k+1} = \arg \min L(\alpha_i, \phi_i^k, \lambda_i^k)$$

Cloud:

$$\phi_i^{k+1} = \arg \min L(\alpha_i^{k+1}, \phi_i, \lambda_i^k)$$

$$\lambda_i^{k+1} = \arg \min L(\alpha_i^{k+1}, \phi_i^{k+1}, \lambda_i)$$

end for

为确定最优的协作规模, 本文采用广度优先搜索算法对参与协作的边缘服务器进行组织。生成树以边缘服务器 ξ_i 为根, 按照广度优先搜索算法, 将边缘服务器 ξ_i 的邻居加入此生成树, 邻居重复上述过程, 直到生成树中的节点数量等于 $M \times \phi_i^k$ 。需要注意的是, 本文组织可以协作卸载的边缘服务器产生的生成树属于非完全 k 叉树, 其中 k 为边缘服务器平均邻居节点个数。整个 k 叉树构成了协作域, 其中根和叶子节点为协作卸载节点, 中间节点为转发节点。

综上, 边缘服务器 ξ_i 接收到感知数据的卸载请求, 其具体的实现步骤如下。

步骤 1 (系统初始化): 假设边缘服务器个数为 M , 当前有 K 类感知数据需要卸载, 采用随机卸载机制。

步骤 2 当第 k 类感知数据到达边缘服务器 ξ_i , 边缘服务器 ξ_i 运行算法 1 获得最优的数据卸载比例 α_i^k 和参与协作边缘服务器的协作比例 ϕ_i^k 。

步骤 3 当协作节点的规模确定之后, 边缘服务器 ξ_i 采用广度优先搜索算法组织协作边缘服务器。

步骤 4 云数据中心处理剩余 $(1 - \alpha_i^k)$ 部分的感知数据, 并将结果返还给用户。

4 仿真结果

本文基于 MATLAB 开发一个协同边缘计算仿真系统测试分布式 ADMM。系统的初始设置为： $M=100$ 、 $E=800$ 、 $D=8$ 、 $\tau_{ki}=0.03$ 、 $\tau_{ij}=0.2$ 、 $\tau_{ic}=0.4$ 。本文假设 λ_i 为边缘服务器 ξ_i 通信范围内的任务负载到达率，令 L_k 表示布置第 k 类应用的边缘服务器的数量。此外，本文还将分布式 ADMM 与其他几种算法进行对比，以评估其时延和能量消耗。对比算法分别是文献[22]针对工作负载转发问题的分布式优化算法 (DOA, distributed optimization algorithm)、文献[30]针对协同边缘计算网络公平性的公平合作算法 (FCA, fairness cooperation algorithm) 和文献[31]考虑邻居节点协作的多子任务到多服务器卸载方案 (MTMS, multi-subtasks-to-multi-servers offloading scheme)。

首先，讨论分布式 ADMM 算法在初始参数设置下的收敛性能，分布式 ADMM 中变量随迭代次数的变化如图 2 所示，可以看出拉格朗日乘子 λ 、卸载比例 α 和协作比例 ϕ ，这 3 个参数可以在较少的迭代 (小于 5 次迭代) 实现收敛。

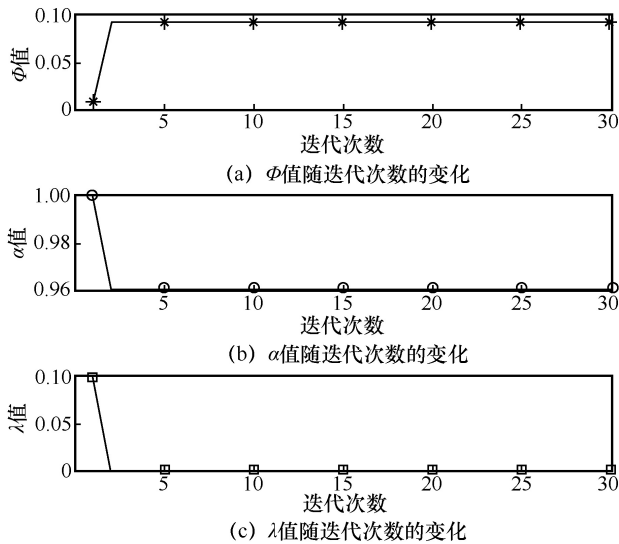


图 2 分布式 ADMM 中变量随迭代次数的变化

协作比例与时延、能耗阈值的关系如图 3 所示，可以看出随着系统时延和能耗的增加，协作比例随之增加，这是因为随着时延和能耗阈值的增加，可以参与协作的边缘服务器的数量也会随之增加，故协作比例与时延和能耗的关系呈正相关。

卸载比例与时延、能耗阈值的关系如图 4 所示，

可以看出随着系统时延和能耗的增加，卸载比例的值会逐渐提高，并趋于稳定状态，这是因为边缘服务器受限于自身硬件能力，算力资源有限，对感知数据的处理能力不会随着时延和能耗阈值的增加而无限提高，故最终趋于稳定状态。

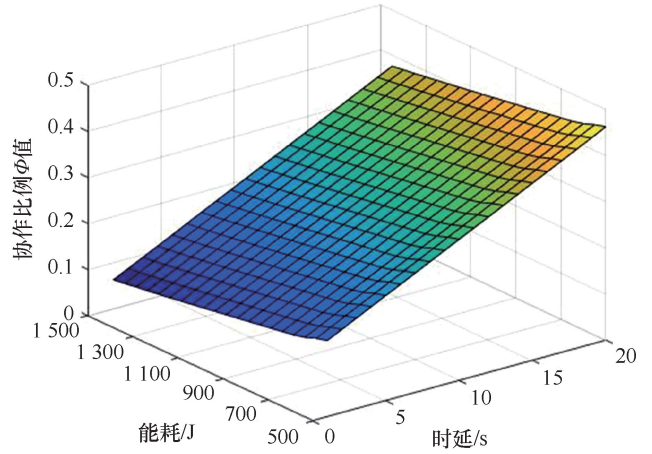


图 3 协作比例与时延、能耗阈值的关系

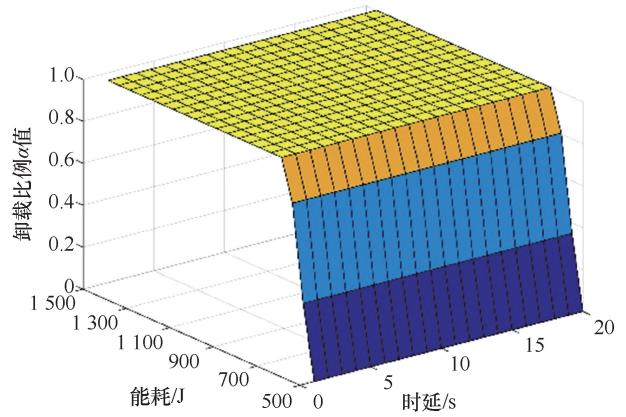


图 4 卸载比例与时延、能耗阈值的关系

协作卸载转发树如图 5 所示，其中黑色的点表示终端设备，红色的三角形表示边缘服务器，绿色的点表示随机选取的终端设备，它将数据处理请求发送给邻近的边缘服务器，如果邻近的边缘服务器无法处理数据请求，则将数据请求转发给其他边缘服务器进行协同处理。蓝色的标记表示协作处理感知数据的边缘服务器，所有的边缘服务器和物联网感知设备都在 $120\text{ m} \times 120\text{ m}$ 的区域内遵循泊松簇分布，当协作规模确定之后，沿着图中所示的转发树查找可以协作处理的边缘服务器，其中生成树的规模为 9。

$L_k=5$ 、 $L_k=10$ ，时延与能耗对比分别如图 6、图 7 所示，通过对比可以得出分布式 ADMM 在

时延和能耗的取值小于 FCA、DOA 和 MTMS，并且 FCA 相对于没有考虑系统公平性的 DOA 均有较小的时延和能耗，MTMS 相对于 FCA 在时延和能耗上有较大的优化。当 $L_k=5$ ，分布式 ADMM 相对于 MTMS，在时延和能耗上分别平均降低 11%和 22%。当 $L_k=10$ ，分布式 ADMM 相对于 MTMS 在时延和能耗上分别平均降低了 15%和 42%。这是因为所提算法在提高感知数据卸载比例的同时，保持较小的协作代价，在系统时延和能耗的约束下尽可能提高感知数据的卸载比例，同时对协作规模进行约束，避免边缘服务器协作规模增大产生额外的网络开销。因此所提分布式 ADMM 相对于没有考虑协作规模的 FCA、DOA 和 MTMS，在系统时延和能耗上均有较大的优化。

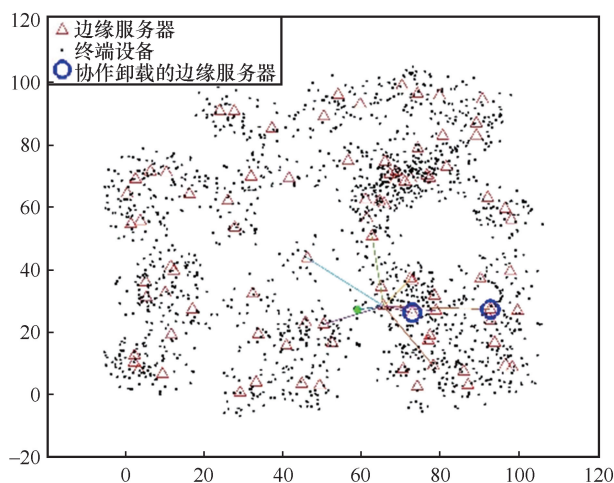


图5 协作卸载转发树

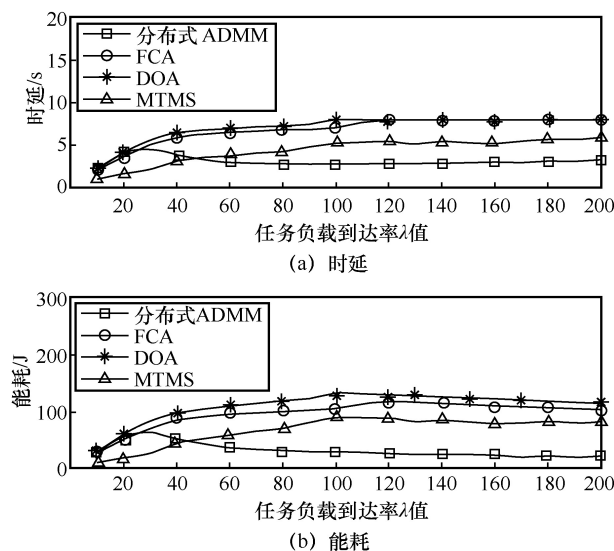


图6 $L_k=5$ ，时延与能耗对比

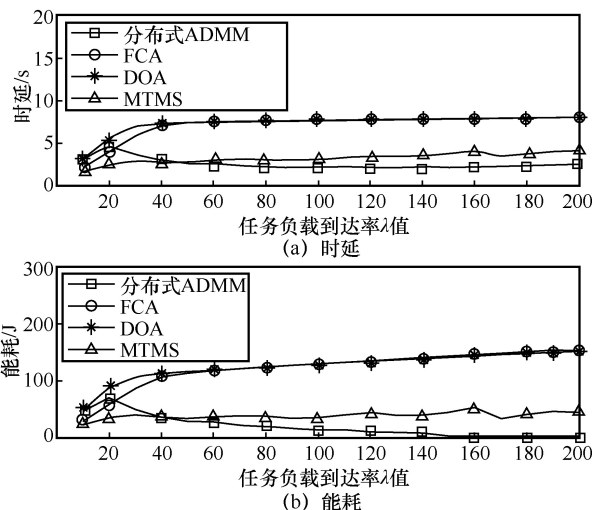


图7 $L_k=10$ ，时延与能耗对比

5 结束语

本文研究了协同边缘计算场景下感知数据的协作卸载问题。本文的目标是在网络时延和系统能耗的约束下，在提高感知数据的卸载比例 α_i^k 的同时，确定最优的协作规模 ϕ_i^k ，保持较小的协作代价并提高成功查找协作节点的概率。同时，本文提出了在网络时延和系统能耗的约束下，让感知数据卸载比例与协作比例的和最大的问题，并采用分布式 ADMM 给出了问题的最优解。本文将所提解决方案与其他算法进行对比，仿真结果表明，分布式 ADMM 相对于 FCA、DOA 和 MTMS 在时延和能耗上均有较大优化。在今后的工作中，笔者会进一步考虑将边缘服务器的性能差异和资源分配应用到所提算法与模型中。

参考文献:

- [1] PAN J L, MCELHANNON J. Future edge cloud and edge computing for Internet of Things applications[J]. IEEE Internet of Things Journal, 2018, 5(1): 439-449.
- [2] HU Y C, PATEL M, SABELLA D, et al. Mobile edge computing—a key technology towards 5G[J]. ETSI white paper, 2015, 11(11): 1-16.
- [3] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.
- [4] CHEN M Z, CHALLITA U, SAAD W, et al. Artificial neural networks-based machine learning for wireless networks: a tutorial[EB]. 2017.
- [5] DENG S G, XIANG Z Z, TAHERI J, et al. Optimal application deployment in resource constrained distributed edges[J]. IEEE Transactions on Mobile Computing, 2021, 20(5): 1907-1923.
- [6] ZHAO H L, DENG S G, LIU Z J, et al. Distributed redundancy scheduling for microservice-based applications at the edge[J]. 2021 IEEE World Congress on Services (SERVICES), 2021: 1.
- [7] DU J B, ZHAO L Q, FENG J, et al. Computation offloading and resource allocation in mixed fog/cloud computing systems with Min-max fairness guarantee[J]. IEEE Transactions on Communications, 2018, 66(4): 1594-1608.

- [8] CHEN M H, LIANG B, DONG M. Multi-user multi-task offloading and resource allocation in mobile cloud systems[J]. IEEE Transactions on Wireless Communications, 2018, 17(10): 6790-6805.
- [9] CHEN M H, DONG M, LIANG B. Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints[J]. IEEE Transactions on Mobile Computing, 2018, 17(12): 2868-2881.
- [10] DU J B, ZHAO L Q, CHU X L, et al. Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems[J]. IEEE Transactions on Vehicular Technology, 2019, 68(2): 1757-1771.
- [11] LIU C F, BENNIS M, DEBBAH M, et al. Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing[J]. IEEE Transactions on Communications, 2019, 67(6): 4132-4150.
- [12] XING H, LIU L, XU J, et al. Joint task assignment and resource allocation for D2D-enabled mobile-edge computing[J]. IEEE Transactions on Communications, 2019, 67(6): 4193-4207.
- [13] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856-868.
- [14] WANG K, YIN H, QUAN W, et al. Enabling collaborative edge computing for software defined vehicular networks[J]. IEEE Network, 2018, 32(5): 112-117.
- [15] SAHNI Y, CAO J N, YANG L, et al. Multi-hop multi-task partial computation offloading in collaborative edge computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(5): 1133-1145.
- [16] DENG S G, ZHANG C, LI C, et al. Burst load evacuation based on dispatching and scheduling in distributed edge networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(8): 1918-1932.
- [17] SAEED A, AMMAR M, HARRAS K A, et al. Vision: the case for symbiosis in the Internet of Things[C]/MCS '15: Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services. 2015: 23-27.
- [18] TRAN T X, HAJISAMI A, PANDEY P, et al. Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges[J]. IEEE Communications Magazine, 2017, 55(4): 54-61.
- [19] CHEN L X, XU J. Socially trusted collaborative edge computing in ultra dense networks[C]/SEC '17: Proceedings of the Second ACM/IEEE Symposium on Edge Computing. 2017: 1-11.
- [20] CHI G X, WANG Y M, LIU X, et al. Latency-optimal task offloading for mobile-edge computing system in 5G heterogeneous networks[C]/Proceedings of 2018 IEEE 87th Vehicular Technology Conference. Piscataway: IEEE Press, 2018: 1-5.
- [21] REN J K, YU G D, CAI Y L, et al. Latency optimization for resource allocation in mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(8): 5506-5519.
- [22] XIAO Y, KRUNZ M. QoE and power efficiency tradeoff for fog computing networks with fog node cooperation[C]/Proceedings of IEEE INFOCOM 2017 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2017: 1-9.
- [23] YUAN P Y, et al. Caching hit ratio maximization in mobile edge computing with node cooperation[J]. Computer Networks, 2021, 200: 108507.
- [24] WANG Q, CHEN S G. Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks[J]. Transactions on Emerging Telecommunications Technologies, 2020, 31(12): e3880.
- [25] XING H, LIU L, XU J, et al. Joint task assignment and wireless resource allocation for cooperative mobile-edge computing[C]/Proceedings of 2018 IEEE International Conference on Communications. Piscataway: IEEE Press, 2018: 1-6.
- [26] CHEN M, HAO Y X. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587-597.
- [27] VU T T, NGUYEN D N, HOANG D T, et al. Optimal energy efficiency with delay constraints for multi-layer cooperative fog computing networks[J]. IEEE Transactions on Communications, 2021, 69(6): 3911-3929.
- [28] HUANG X G, CUI Y F, CHEN Q B, et al. Joint task offloading and

QoS-aware resource allocation in fog-enabled Internet-of-things networks[J]. IEEE Internet of Things Journal, 2020, 7(8): 7194-7206.

- [29] LAN X L, CAI L, CHEN Q C. Execution latency and energy consumption tradeoff in mobile-edge computing systems[C]/Proceedings of 2019 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE Press, 2019: 123-128.

- [30] DONG Y F, GUO S T, LIU J D, et al. Energy-efficient fair cooperation fog computing in mobile edge networks for smart city[J]. IEEE Internet of Things Journal, 2019, 6(5): 7543-7554.

- [31] WANG J, WU W B, LIAO Z F, et al. An energy-efficient off-loading scheme for low latency in collaborative edge computing[J]. IEEE Access, 2019, 7: 149182-149190.

[作者简介]



袁培燕（1978—），男，博士，河南师范大学教授，主要研究方向为边缘计算与群智感知、移动自组织与机会网络、网络大数据等。



邵赛珂（1998—），男，河南师范大学硕士生，主要研究方向为移动边缘计算。



魏然（1978—），男，河南师范大学工程师，主要研究方向为数据库与软件开发。



张俊娜（1979—），女，博士，河南师范大学副教授，主要研究方向为移动边缘计算、服务计算等。



赵晓焱（1981—），女，博士，河南师范大学副教授，主要研究方向为移动边缘计算、D2D通信、物联网等。