

低地球轨道卫星边缘计算场景中任务卸载与 资源分配联合优化算法

夏玮玮, 胡静, 宋铁成

(东南大学移动通信全国重点实验室, 江苏 南京 210096)

摘要: 针对低地球轨道 (LEO) 卫星边缘计算场景中地面用户计算任务的卸载需求, 提出联合卸载与资源分配优化 (JORAO) 算法。考虑到 LEO 卫星的有限覆盖时间, 以最小化所有地面用户的平均服务时延为目标, 联合优化卸载策略、LEO 卫星的通信和计算资源分配。将任务卸载与资源分配的联合优化问题分解为卸载决策和资源分配子问题, 使用交替优化方法, 获得原始优化问题的次优解。对于任务卸载决策子问题, 将其建模为联盟博弈模型, 当博弈达到纳什均衡时, 获得最小化平均服务时延的地面用户卸载策略; 对于资源分配子问题, 使用拉格朗日乘子法获得最优的通信和计算资源分配结果。此外, 还证明了所提算法的收敛性和稳定性。仿真结果表明, 所提算法具有良好的收敛性, 能显著降低地面用户的平均服务时延和提高任务卸载成功率。

关键词: 低地球轨道卫星; 边缘计算; 卸载; 资源分配; 联盟博弈

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024135

Joint optimization algorithm for task offloading and resource allocation in low earth orbit satellites edge computing scenario

XIA Weiwei, HU Jing, SONG Tiecheng

National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China

Abstract: Aiming at the offloading requirements of ground users' computing tasks in edge computing scenario of low earth orbit (LEO) satellites, a joint offloading and resource allocation optimization (JORAO) algorithm was proposed. Considering the limited coverage time of LEO satellites, the offloading strategy, the allocation of communication and computing resources of LEO satellites were jointly optimized to minimize the average service delay of all ground users. The joint optimization problem of task offloading and resource allocation was decomposed into offloading decision and resource allocation sub-problems, and an alternating optimization method was used to obtain the suboptimal solution of the original optimization problem. The task offloading decision sub-problem was modeled as a coalition game model, and when the game reached Nash equilibrium, the ground user offloading strategy that minimized the system delay was obtained. For the resource allocation sub-problem, the Lagrange multiplier method was used to obtain the optimal bandwidth and compute resource allocation results. Moreover, the convergence and stability of the proposed algorithm were also demonstrated. The simulation results show that the proposed algorithm has excellent convergence and can significantly reduce the average service delay of ground users, as well as improve the task offloading success rate.

Keywords: low earth orbit satellite, edge computing, offloading, resource allocation, coalition game

收稿日期: 2024-03-12; 修回日期: 2024-06-12

基金项目: 江苏省重点研发计划基金资助项目(No.BE2020084-2)

Foundation Item: The Key Research and Development Program of Jiangsu Province (No.BE2020084-2)

0 引言

随着物联网的快速发展,越来越多的用户终端被布置在偏远地区。这些区域由于地理环境的限制,几乎没有支持5G移动通信技术的蜂窝网络等地面网络覆盖,因此,用户终端难以通过地面网络获得相应的服务。此外,在发生极端自然灾害等场景下,5G蜂窝网络无法提供网络的完全覆盖和对移动终端业务的有效支持,很难满足应急场景中各类型用户终端随时通信的需求^[1]。低地球轨道(LEO, low earth orbit)卫星互联网以其高带宽、广覆盖的优势,可实现通信网络的灵活扩展,满足偏远地区和应急场景中用户终端业务的接入。目前,许多国家已经开始了LEO卫星互联网项目,比如OneWeb^[2]、Starlink^[3]以及O3b^[4]等,我国对LEO卫星互联网也正在进行规划和部署。

偏远地区或应急场景中具有计算密集型和时延敏感型应用的用户终端,需要将任务卸载到中心云服务器或布置在网络边缘(如接入点和基站)的移动边缘计算(MEC, mobile edge computing)节点,从而有效解决其计算能力薄弱、存储空间不足、时延过大等问题^[5]。这些场景中用户终端的计算任务可以通过LEO卫星互联网转发到中心云服务器处理。然而,用户终端的计算任务通过LEO卫星互联网转发至中心云服务器的传输时延很难满足业务的实时性要求。比如,OneWeb卫星支持的端到端往返时延约为50 ms,Starlink卫星支持的端到端往返时延约为40 ms,而O3b卫星支持的端到端往返时延约为200 ms。由于LEO卫星在低地球轨道运行,将MEC服务器装载至LEO卫星处理计算任务,则可省去LEO卫星至中心云服务器的数据传输时延,从而大大降低了任务的总服务时延^[6]。鉴于LEO卫星互联网的广覆盖和低时延特性,将边缘计算技术引入LEO卫星互联网,利用LEO天基计算资源,向地面用户终端,特别是偏远地区或应急场景中的用户终端提供低时延的计算服务成为重要的解决方案^[6-8]。

近几年,基于LEO卫星的MEC引起了学术界和工业界的重视,被认为是未来网络发展的重要研究方向^[6-10]。文献[11]将LEO卫星MEC场景中的服务请求分配和处理问题建模为混合整数线性规划的优化问题,从而最小化所有用户从LEO-MEC系统获得服务的整体代价,同时最大化获得服务的用户

比例。尽管该文对每颗LEO卫星的覆盖区域进行了划分,然而在地面用户选择可卸载计算任务的LEO卫星时,并没有考虑到每颗LEO卫星覆盖范围的变化。文献[12]以最小化终端的响应时延和能耗成本为目标,提出了基于博弈论的方法以优化卫星边缘计算场景中地面终端的卸载策略。但是该文假设地面用户只分布在很小的固定区域,所有地面用户与LEO卫星的连接关系都是相同的。文献[13]研究了LEO和无人机(UAV, unmanned aerial vehicle)辅助的自组网中的计算资源定价以及任务卸载策略,将LEO-MEC服务器以及UAV-MEC服务器与边缘节点之间的交互建模为斯坦伯格(Stackelberg)博弈模型,从而降低任务处理时延和系统能耗。但是该文仅考虑了单颗LEO卫星覆盖场景,并没有考虑到LEO卫星移动导致的覆盖范围变化。文献[14]研究了具有3层计算架构的混合中心云和边缘计算LEO卫星网络中的卸载决策以最小化所有地面用户的总能耗,使用交替方向乘子法(ADMM, alternating direction method of multiplier)获得原始优化问题的次优解。尽管该文考虑了LEO卫星的覆盖时间限制,但是向地面用户固定分配通信和计算资源。文献[11-14]都只考虑了LEO卫星边缘计算场景中的卸载决策问题,并没有联合优化计算和通信资源分配。

文献[9]以最小化所有地面物联网终端的总时延同时满足能耗、计算和缓存约束为目标,提出了LEO卫星边缘计算场景中的计算卸载和资源分配联合优化策略,并使用拉格朗日对偶分解法获得闭式最优解。然而,该文没有考虑LEO卫星的有限覆盖时间,并且算法复杂度较高。文献[15]提出了一个新颖的星地物联网场景中的MEC框架,将地面物联网终端向LEO卫星的卸载过程分成空间和地面2个阶段,并在此基础上提出能量有效的计算卸载和资源分配算法。在该框架中,地面卫星网关作为地面物联网终端的接入点与LEO卫星通信,并将地面物联网终端的计算任务同时卸载至多颗LEO卫星。然而,该卸载过程并没有考虑LEO卫星的移动。文献[16]以最大化系统效用为目标,提出了星地协作网络中的任务卸载和资源分配联合优化算法,实现了多服务器多用户场景中的任务卸载决策及计算和通信资源联合分配。在该场景中,用户可以将计算任务直接卸载至LEO边缘计算卫星或者

通过地球同步轨道卫星将任务卸载至云中心。然而,该文在建立优化模型时并没有考虑 LEO 卫星的覆盖时间以及 LEO 卫星-地面用户链路存在的有效时间。

不同于位于地面的 MEC 系统, LEO 卫星的位置是动态变化的。只有当地面用户和 LEO 卫星之间的相对位置满足一定的关系时,地面用户才能向 LEO 卫星进行计算任务卸载。因此, LEO 卫星有限的覆盖时间和链路存在时间会影响地面用户与该卫星的通信可靠性,从而导致计算任务卸载过程的间歇中断,降低用户的服务体验质量(QoE, quality of experience)。所以,有别于地面边缘计算场景, LEO 卫星边缘计算场景中的任务卸载与资源分配必须考虑上述限制因素,从而建立不同的系统优化模型。不同于已有文献的研究,本文考虑一种 LEO 卫星边缘计算场景,该场景中的地面用户分布于不同 LEO 卫星的覆盖范围内,它们在本地处理计算任务或者将计算任务卸载至覆盖该用户的 LEO 卫星处理。本文的主要贡献如下。

1) 提出一种 LEO 卫星边缘计算场景中的联合卸载与资源分配优化(JORAO, joint offloading and resource allocation optimization)算法,以最小化所有地面用户的平均服务时延为目标,联合优化面向地面用户的卸载决策、通信和计算资源分配。

2) 建立任务卸载与资源分配的联合优化模型,考虑 LEO 卫星的有限覆盖时间,以及 LEO 卫星与地面用户之间链路的暂态性,以地面用户与 LEO 卫星之间的最长通信时间为约束,将原始优化问题分解为卸载决策和资源分配子问题,使用交替优化方法,获得原始优化问题的次优解。

3) 任务卸载决策子问题被建模为联盟博弈模型,当博弈达到纳什均衡时,获得最小化平均服务时延的所有地面用户的卸载策略。卸载决策包括是将计算任务在本地进行计算,还是卸载至 LEO 卫星以及卸载至哪颗 LEO 卫星。对于资源分配子问题,使用拉格朗日乘子法和 KKT(Karush-Kuhn-Tucker)条件求解获得通信和计算资源分配结果。

4) 仿真结果表明,与已有算法相比,所提算法具有良好的收敛性和较短的运行时间。当地面用户数目、LEO 卫星数目和计算任务长度改变时,所提算法能显著降低地面用户的平均服务时延。随着 LEO 卫星与地面用户之间的俯仰角改变,所提算法

的平均服务时延和任务卸载成功率显著优于已有算法。

1 系统模型

本文研究的 LEO 卫星边缘计算场景如图 1 所示,其中, LEO 卫星运行在轨道中,地面用户终端位于没有地面公共移动通信网络支持的偏远地区。假设场景中包括 N 个 LEO 卫星和 M 个用户终端。记 LEO 卫星的集合为 $\mathcal{N} = \{1, 2, \dots, N\}$, 所有用户终端的集合为 $\mathcal{M} = \{1, 2, \dots, M\}$ 。由于 LEO 卫星边缘计算场景能够通过 LEO 卫星上部署 MEC 服务器向地面用户终端提供全域覆盖的在轨计算服务,因此假设每个 LEO 卫星都装载了一台 MEC 服务器^[9,17]。考虑到 LEO 卫星是载荷有限系统, MEC 服务器可采用诸如 Docker 的轻量级管理平台^[18], 减小其对卫星能耗以及负载的影响。用户的任务用二元组 $\langle c_m, l_m \rangle_{m \in \mathcal{M}}$ 表示,其中, c_m 为处理任务所需的 CPU 周期数, l_m 为任务大小的比特数。根据用户终端的不同需求,其计算任务可由用户终端自身处理,也可以通过无线链路卸载到 LEO 卫星进行处理。此外,假设每个用户只有一个计算任务将被处理,并且该计算任务不能被分离^[19]。考虑到 LEO 卫星具有高度移动性,地面用户终端与 LEO 卫星之间的通信时间受限于 LEO 卫星的覆盖时间。与文献[20-21]类似,本文在 LEO 卫星边缘计算场景中基于软件定义网络(SDN, software define network)和网络功能虚拟化技术实现全局控制,而执行所提算法的 SDN 服务器可以部署在 LEO 卫星或网关上。

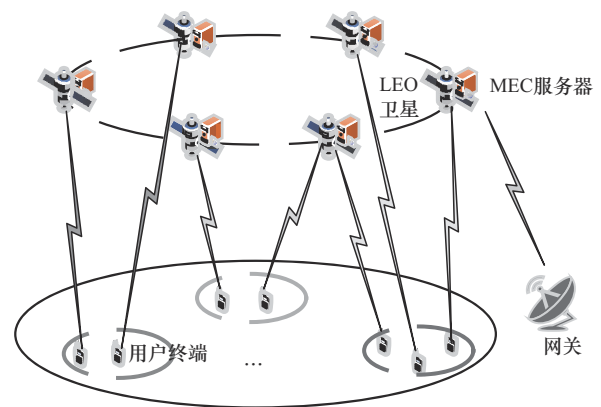


图1 本文研究的 LEO 卫星边缘计算场景

1.1 覆盖时间模型

根据文献[12], LEO 卫星和地面用户之间的位

置关系如图2所示。假设 d_{nm} 表示地面用户 m 与卫星轨道间的高度, R_c 表示地球的半径, s_{nm} 表示地面用户 m 与LEO卫星 n 之间的距离, θ_{nm} 表示地面用户 m 与LEO卫星 n 之间的俯仰角, 并随着卫星持续移动而变化。假设 α_{nm} 表示LEO卫星 n 与穿过地心垂直于参照线的直线之间的夹角, γ_{nm} 表示与LEO卫星 n 覆盖区域相对应的地心角, 当LEO卫星在参照线以下时, 无法为用户提供服务。

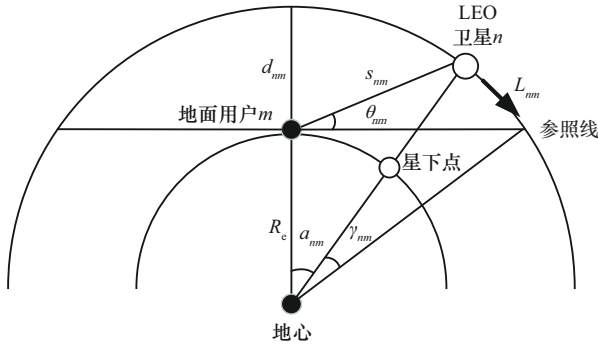


图2 LEO卫星和地面用户之间的位置关系

根据几何关系, θ_{nm} 和 γ_{nm} 可以表示为

$$\theta_{nm} = \arccos\left(\frac{R_c + d_{nm}}{s_{nm}} \sin \alpha_{nm}\right) \quad (1)$$

$$\gamma_{nm} = \arccos\left(\frac{R_c}{R_c + d_{nm}}\right) - \alpha_{nm} \quad (2)$$

地面用户 m 与LEO卫星 n 之间的最长通信时间表示为

$$T_{nm}^{\max} = \frac{L_{nm}}{v_n} \quad (3)$$

其中, v_n 是LEO卫星 n 的速度, L_{nm} 是LEO卫星 n 对地面用户 m 的覆盖弧长, 计算式为

$$L_{nm} = (R_c + d_{nm})\gamma_{nm} \quad (4)$$

1.2 通信模型

当地面用户有计算任务时, 用户选择在本地计算或者卸载至LEO卫星, 以满足其服务质量要求。一般来说, 计算结果相对于原始任务数据量较小, 因此, 将计算结果从卫星传送给用户的时间往往被忽略。用户 m 和卫星 n 之间的上行链路数据传输速率计算式为

$$R_{nm} = B_{nm} \log\left(1 + \frac{h_{nm} p_m}{\sigma_0^2}\right) \quad (5)$$

其中, p_m 是用户 m 的传输功率; σ_0^2 是加性白高斯噪声的功率; h_{nm} 是用户与LEO卫星之间的信道增益, 与自由空间路径损耗和降雨衰减等因素有

关^[22], 由于地面用户与卫星之间链路的降雨衰减可以通过功率控制等方法补偿^[23], 本文不考虑降雨衰减对卫星信号传输质量造成的影响, 此外, LEO卫星运动引起的多普勒频移也可以由地面接收端估计并补偿^[24], 因此, 为了方便分析, 本文也不考虑多普勒频移带来的影响; B_{nm} 是LEO卫星 n 分配给地面用户 m 的带宽资源。使用矩阵 $\mathbf{B} = \langle B_{nm} \rangle_{m \in \mathcal{M}, n \in \mathcal{N}}$ 表示带宽资源的分配策略, 则有

$$\sum_{m=1}^M B_{nm} \leq \bar{W}_n, \forall n \in \mathcal{N} \quad (6)$$

其中, \bar{W}_n 是LEO卫星 n 的总带宽。于是可以得到每个用户的数据传输时延为

$$\tau_{nm}^T = \frac{l_m}{R_{nm}}, \forall m \in \mathcal{M} \quad (7)$$

当用户 m 卸载计算任务至LEO卫星 n 时, 由于卫星与用户之间的物理距离较远, 信号传播时延是不能忽略的, 可表示为

$$\tau_{nm}^P = \frac{s_{nm}}{c}, \forall m \in \mathcal{M} \quad (8)$$

其中, c 是光速。

1.3 卸载模型

假设 $\mathbf{X} = \langle x_{nm} \rangle_{n \in \mathcal{N}, m \in \mathcal{M}}$ 表示卸载策略矩阵, 其中, $x_{nm} \in \{0, 1\}$, $x_{nm} = 1$ 表示地面用户 m 将任务卸载至LEO卫星 n 进行计算, $\sum_{n=1}^N x_{nm} = 0$ 表示用户在本本地处理该任务。由于每个地面用户或者在本本地计算任务或者将任务卸载到某LEO卫星处理, 因此 x_{nm} 满足

$$\sum_{n=1}^N x_{nm} \leq 1, \forall m \in \mathcal{M} \quad (9)$$

1.4 计算模型

本文使用频分复用的方法以减小卫星之间的干扰, 假设每个LEO卫星在不同的频段向用户提供计算服务。由于不同用户的任务有可能被卸载到同一LEO卫星进行计算, 因此需要对LEO卫星的计算资源进行分配。用 \bar{f}_n ($\forall n \in \mathcal{N}$)表示LEO卫星 n 的计算能力, f_{nm} 表示LEO卫星 n 分配给用户 m 的计算资源, 则有

$$\sum_{m=1}^M f_{nm} \leq \bar{f}_n, \forall n \in \mathcal{N} \quad (10)$$

因此, 所有LEO卫星的计算资源分配策略为矩阵 $\mathbf{F} = \langle f_{nm} \rangle_{n \in \mathcal{N}, m \in \mathcal{M}}$ 。

LEO 卫星 n 对用户 m 的任务计算时延表示为

$$\tau_{nm}^c = \frac{c_m}{f_{nm}}, \forall m \in \mathcal{M} \quad (11)$$

用 f_m^l 表示地面用户 m 的 CPU 计算能力, 那么任务的本地计算时间为

$$\tau_m^l = \frac{c_m}{f_m^l} \quad (12)$$

2 优化问题

当用户将任务卸载到 LEO 卫星时, 用户的服务时延中包括了数据传输时延、信号传播时延和任务计算时延, 则所有用户的平均服务时延为

$$T(\mathbf{X}, \mathbf{F}, \mathbf{B}) = \frac{1}{M} \left[\sum_{n=1}^N x_{nm} (\tau_{nm}^T + \tau_{nm}^P + \tau_{nm}^C) + \left(1 - \sum_{n=1}^N x_{nm}\right) \tau_m^L \right] \quad (13)$$

同时考虑任务卸载决策、计算资源分配和通信资源分配, 以满足在有限资源约束以及 LEO 卫星覆盖时间约束的条件下最小化用户平均服务时延为目标, 可以得到如下优化问题

$$\begin{aligned} & \min_{\mathbf{X}, \mathbf{F}, \mathbf{B}} T(\mathbf{X}, \mathbf{F}, \mathbf{B}) \\ \text{s.t. } & \text{C1: } \tau_{nm}^P + \tau_{nm}^T + \tau_{nm}^C \leq T_{nm}^{\max}, \quad m \in \mathcal{M}, n \in \mathcal{N} \\ & \text{C2: } x_{nm} \in \{0, 1\}, \quad m \in \mathcal{M}, n \in \mathcal{N} \\ & \text{C3: } \sum_{n=1}^N x_{nm} \leq 1, \quad m \in \mathcal{M} \\ & \text{C4: } f_{nm} \geq 0, \quad n \in \mathcal{N}, m \in \mathcal{M} \\ & \text{C5: } \sum_{m=1}^M f_{nm} \leq \bar{f}_n, \quad n \in \mathcal{N} \\ & \text{C6: } B_{nm} \geq 0, \quad n \in \mathcal{N}, m \in \mathcal{M} \\ & \text{C7: } \sum_{m=1}^M B_{nm} \leq \bar{W}_n, \quad n \in \mathcal{N} \end{aligned} \quad (14)$$

式(14)所示优化问题是一个混合整数非线性规划问题, 其中, C1 为卫星覆盖时间约束, C2 和 C3 为卸载决策变量的约束条件, C4 和 C5 为计算资源约束, C6 和 C7 为通信资源约束。

3 联合卸载与资源分配优化算法

3.1 问题分解

从式(14)可以看出, 优化问题同时受连续变量 \mathbf{F} 和 \mathbf{B} 以及离散变量 \mathbf{X} 的影响, 而且具有非凸性, 因此该问题是 NP-Hard 问题, 难以求得全局最优解。LEO 卫星的持续移动使得地面用户与 LEO 卫星的链路连接状况动态变化, 而使用传统的优化方法获得该动态决策问题的最优解是非常具有挑战性

的。所以本文考虑将原问题分解为卸载决策子问题和资源分配子问题来寻求其次优解。此外, 所提算法应具有低复杂度, 从而在其迭代稳定之前 LEO 卫星的覆盖条件不发生变化。因此, 先通过联盟博弈获得本地计算或卸载任务至 LEO 卫星, 以及地面用户与 LEO 卫星之间的联盟关系决策 \mathbf{X}^* ; 再使用交替优化方法, 进一步将优化问题解耦成计算资源和通信资源分配子问题。为降低算法复杂度, 在求解计算资源和通信资源分配子问题时, 将全局优化问题分解为若干个局部优化子问题, 分别求得计算资源分配结果 \mathbf{F}^* 和通信资源分配结果 \mathbf{B}^* , 进而得到原问题的次优解 $\langle \mathbf{X}^*, \mathbf{F}^*, \mathbf{B}^* \rangle$ 。

3.2 联盟博弈模型

地面用户都被视为博弈参与者, 它们需要通过任务卸载决策来决定在本地计算或卸载任务至 LEO 卫星。下面给出联盟博弈模型 $\{\mathcal{M}, \mathcal{R}, \mathcal{Q}\}$ 的描述。

参与者。集合 \mathcal{M} 中的所有地面用户。

联盟。联盟集合为 $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{M+N}\}$ 。由于存在 M 个地面用户和 N 个 LEO 卫星, 并且考虑二进制卸载决策, 因此最多可以形成 $M+N$ 个联盟。在联盟集合 \mathcal{Q} 中, 对于任意的 $k \neq l$ 且 $1 \leq k, l \leq M+N$, 有 $\mathcal{Q}_k \cap \mathcal{Q}_l = \emptyset$, 并且 $\bigcup_{k=1}^{M+N} \mathcal{Q}_k = \mathcal{M}$ 。当 $1 \leq k \leq N$ 时, 联盟 \mathcal{Q}_k 表示将任务卸载到 LEO 卫星 k 的地面用户集合。考虑到 LEO 卫星的移动性, 地面用户的计算任务只能被卸载至覆盖到该用户的 LEO 卫星, 因此联盟 \mathcal{Q}_k 中的地面用户必须满足 $\tau_{km}^P + \tau_{km}^T + \tau_{km}^C \leq T_{km}^{\max}$, $m \in \mathcal{Q}_k$ 。当 $N+1 \leq k \leq M+N$ 时, 联盟 \mathcal{Q}_k 表示在本地执行计算任务的地面用户, 每个联盟 \mathcal{Q}_k 只包含了唯一的元素。

策略。每个参与者的策略是根据它在每个联盟的效用来决定是否卸载以及用于任务卸载的 LEO 卫星。

效用函数。联盟 \mathcal{Q}_k 的效用函数表示为 $\mathcal{R}(\mathcal{Q}_k)$, 当 $1 \leq k \leq N$ 时, $\mathcal{R}(\mathcal{Q}_k)$ 是联盟中所有利用 LEO 卫星 k 进行任务卸载的地面用户可以获得的总卸载收益; 当 $N+1 \leq k \leq M+N$ 时, $\mathcal{R}(\mathcal{Q}_k)$ 是地面用户在本地计算任务时的收益。

定义 1 联盟 \mathcal{Q}_k 的效用函数 $\mathcal{R}(\mathcal{Q}_k)$ 是非负实数, 且表示联盟 \mathcal{Q}_k 的总收益。

当地面用户的计算任务被卸载至 LEO 卫星时, 只有卸载时延低于本地计算时延, 用户才能因卸载

而获得收益。因此,当 $1 \leq k \leq N$ 时, \mathcal{Q}_k 的效用函数为

$$\mathcal{R}(\mathcal{Q}_k) = \sum_{m \in \mathcal{Q}_k} \mathcal{R}_m(\mathcal{Q}_k) = \sum_{m \in \mathcal{Q}_k} (\tau_m^L - (\tau_{km}^P + \tau_{km}^T + \tau_{km}^C)) \quad (15)$$

当 $N+1 \leq k \leq M+N$ 时,即地面用户在本地计算任务,此时联盟效用 $\mathcal{R}(\mathcal{Q}_k) = 0$ 。

在形成联盟的过程中,每个地面用户会采用一定的偏好关系来比较任意2个联盟。偏好关系如定义2所示。

定义2 对于地面用户 m ,其偏好关系 $>_m$ 定义为其可能形成的所有联盟集合上完整的、自反的和可传递的二进制关系^[25]。 $\mathcal{Q}_1 >_m \mathcal{Q}_2$ 表示地面用户 m 严格地倾向于成为联盟 \mathcal{Q}_1 的成员。

博弈中的每个地面用户都可以根据其偏好关系和收益来决定加入或离开某特定联盟。对于任意的地面用户 $m \in \mathcal{M}$,以下关系成立

$$\mathcal{Q}_s >_m \mathcal{Q}_k \Leftrightarrow \mathcal{R}(\mathcal{Q}_s) + \mathcal{R}(\mathcal{Q}_k \setminus m) > \mathcal{R}(\mathcal{Q}_s \setminus m) + \mathcal{R}(\mathcal{Q}_k)$$

$$\text{s.t. } \mathcal{R}_j(\mathcal{Q}_i) \geq 0, \mathcal{R}_j(\mathcal{Q}_i \setminus m) \geq 0, \forall j \in \{\mathcal{Q}_i \setminus m\}, i = s, k \quad (16)$$

由式(16)可以看出,若地面用户 m 的加入使联盟 \mathcal{Q}_s 的总收益大于联盟 \mathcal{Q}_k 的总收益,且联盟 \mathcal{Q}_s 和 \mathcal{Q}_k 中没有其他地面用户 j 会因为地面用户 m 的变动收益为负,则地面用户 m 更倾向于加入联盟 \mathcal{Q}_s 。地面用户根据式(16)中的偏好关系从联盟 \mathcal{Q}_k 加入联盟 \mathcal{Q}_s 的过程称为切换。所有地面用户经过切换,最终形成联盟集合,也称为联盟分区。

定义3 对于联盟分区 $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{M+N}\}$,当且仅当 $\forall m \in \mathcal{M}, m \in \mathcal{Q}_s$ 且 $m \notin \mathcal{Q}_k, \mathcal{Q}_s >_m \mathcal{Q}_k \cup \{m\}$ 都成立时, \mathcal{Q} 是纳什稳定的,其中, $k \neq s, 1 \leq k, s \leq M+N$ 。

定义4 对于联盟分区 $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{M+N}\}$,

$$\text{其效用 } \mathcal{R}(\mathcal{Q}) = \sum_{k=1}^{M+N} \mathcal{R}(\mathcal{Q}_k)。$$

3.3 基于联盟博弈的卸载决策

根据联盟的效用函数比较地面用户的偏好关系从而执行切换操作,当没有地面用户改变其当前策略时,博弈达到纳什均衡,获得稳定的联盟分区,最终得到最小化系统时延的所有地面用户的卸载策略 \mathbf{X}^* ,如算法1所示。

算法1 基于联盟博弈的卸载决策

输入 $M, N, \langle c_m, l_m \rangle_{m \in \mathcal{M}}, \langle \bar{f}_n \rangle_{n \in \mathcal{N}}, p_m, f_m^l, h_{nm}, s_{nm},$

$\forall m \in \mathcal{M}, n \in \mathcal{N}$

输出 $\langle \mathbf{X}^*, \mathbf{F}^0, \mathbf{B}^0 \rangle$

初始化 建立地面用户的初始随机联盟分区 \mathcal{Q}_{ini} , 设置 $\mathcal{Q}_{cur} = \mathcal{Q}_{ini}$, $m = 0$, 迭代次数 $r_1 = 0$, 最大迭代次数 r_1^{\max}

1) repeat

2) while $m \leq M$

3) $m = m + 1$

4) 选择地面用户 m 并找到其当前所在的联盟 \mathcal{Q}_k

5) for $\mathcal{Q}_s \neq \mathcal{Q}_k, s \neq k, 1 \leq s, k \leq M+N, \mathcal{Q}_s \in \mathcal{Q}_{cur}$

6) if 满足偏好关系 $\mathcal{Q}_s >_m \mathcal{Q}_k$ then

7) 地面用户 m 执行切换,离开当前联盟 \mathcal{Q}_k ,加入 \mathcal{Q}_s ,即设置 $\mathcal{Q}'_k = \mathcal{Q}_k \setminus m$,
 $\mathcal{Q}'_s = \mathcal{Q}_s \cup m$

8) 更新 $\mathcal{Q}_{cur} = \mathcal{Q}_{cur} - \mathcal{Q}_k + \mathcal{Q}'_k$,

$\mathcal{Q}_{cur} = \mathcal{Q}_{cur} - \mathcal{Q}_s + \mathcal{Q}'_s$,且设置 $\mathcal{Q}_k = \mathcal{Q}'_k$

9) end if

10) end for

11) end while

12) 更新 $r_1 = r_1 + 1$

13) until $r_1 \geq r_1^{\max}$

14) \mathcal{Q}_{cur} 收敛到纳什稳定的分区 \mathcal{Q}^* ,获得卸载策略矩阵 \mathbf{X}^*

15) 根据等分的原则获得初始资源分配结果 $\langle \mathbf{F}^0, \mathbf{B}^0 \rangle$

3.4 计算资源分配子问题

获得卸载策略 \mathbf{X}^* 后,由于约束C2和C3只与 \mathbf{X} 有关,对 \mathbf{F} 和 \mathbf{B} 不构成约束,因此可以从约束条件中去掉,优化问题式(14)可被转化为如下资源分配子问题

$$\begin{aligned} \min_{\mathbf{F}, \mathbf{B}} T_1(\mathbf{F}, \mathbf{B}) &= T(\mathbf{X}^*, \mathbf{F}, \mathbf{B}) \\ \text{s.t. } & \text{C1, C4} \sim \text{C7} \end{aligned} \quad (17)$$

使用交替优化方法,进一步将问题式(17)解耦成计算资源分配子问题和通信资源分配子问题。首先把 \mathbf{B}^0 代入问题式(17),得到计算资源分配子问题

$$\begin{aligned} \min_{\mathbf{F}} T_2(\mathbf{F}) &= T_1(\mathbf{F}, \mathbf{B}^0) \\ \text{s.t. } & \text{C1, C4, C5} \end{aligned} \quad (18)$$

当博弈最终收敛到纳什稳定的联盟分区时,就

获得了卸载策略矩阵 \mathbf{X}^* 。对于选择 LEO 卫星进行任务卸载的地面用户，其所在的联盟包含了选择该 LEO 卫星进行任务卸载的所有地面用户，实际形成了多个地面用户与该 LEO 卫星的配对。而每个 LEO 卫星向其联盟内的地面用户分配计算资源是独立进行的，与该联盟之外的 LEO 卫星和地面用户无关。因此，可将全局优化问题式(18)分解为多个（不超过 N 个）相互独立的局部优化子问题，然后对这些子问题并行求解，即可获得计算资源分配结果 \mathbf{F}^* 。忽略目标函数中的常数项，则这些子问题可表示为

$$\begin{aligned} \min_{f_{nm_1}, \dots, f_{nm_{|Q_n|}}} & \sum_{k=1}^{|Q_n|} \frac{c_{m_k}}{f_{nm_k}}, \quad 1 \leq n \leq N \\ \text{s.t. } \widehat{\mathbf{C}}1: & \tau_{nm_k}^P + \tau_{nm_k}^T + \tau_{nm_k}^C \leq T_{nm_k}^{\max}, 1 \leq k \leq |Q_n| \\ & \widehat{\mathbf{C}}4: f_{nm_k} \geq 0, 1 \leq k \leq |Q_n| \\ & \widehat{\mathbf{C}}5: \sum_{k=1}^{|Q_n|} f_{nm_k} \leq \bar{f}_n \end{aligned} \quad (19)$$

定理 1 优化问题式(19)是凸问题。

证明 令

$$G_f(f_{nm_1}, \dots, f_{nm_{|Q_n|}}) = \sum_{k=1}^{|Q_n|} \frac{c_{m_k}}{f_{nm_k}} \quad (20)$$

求二阶偏导可得

$$\frac{\partial^2 G_f}{\partial f_{m_i} \partial f_{m_j}} = \begin{cases} \frac{2c_i}{f_{m_i}^3}, & i = j \\ 0, & i \neq j \end{cases} \quad (21)$$

因此，式(19)中目标函数的 Hessian 矩阵 $\nabla^2 G_f$ 为正定阵。由于约束条件 $\widehat{\mathbf{C}}1$ 与目标函数有相同的形式，从而 $\widehat{\mathbf{C}}1$ 构成凸集。此外， $\widehat{\mathbf{C}}4$ 和 $\widehat{\mathbf{C}}5$ 均为线性约束条件，因此优化问题式(19)是凸问题，可以利用 KKT 条件来求解。证毕。

优化问题式(19)的拉格朗日函数为

$$\begin{aligned} L_f(\mathbf{F}_n, \boldsymbol{\kappa}, \nu) &= \sum_{k=1}^{|Q_n|} \frac{c_{m_k}}{f_{nm_k}} - \\ & \sum_{k=1}^{|Q_n|} \kappa_k (\tau_{nm_k}^T + \tau_{nm_k}^P + \tau_{nm_k}^C - T_{nm_k}^{\max}) - \\ & \nu \sum_{k=1}^{|Q_n|} (f_{nm_k} - \bar{f}_n) \end{aligned} \quad (22)$$

KKT 条件为

$$\nabla L_f(\mathbf{F}_n^*, \boldsymbol{\kappa}^*, \nu^*) = 0 \quad (23)$$

$$\tau_{nm_k}^T + \tau_{nm_k}^P + \tau_{nm_k}^C - T_{nm_k}^{\max} \leq 0, 1 \leq k \leq |Q_n| \quad (24)$$

$$f_{nm_k}^* - \bar{f}_n \leq 0, 1 \leq k \leq |Q_n| \quad (25)$$

$$\kappa_k^* \geq 0, 1 \leq k \leq |Q_n| \quad (26)$$

$$\nu^* \geq 0 \quad (27)$$

$$f_{nm_k}^* \geq 0, 1 \leq k \leq |Q_n| \quad (28)$$

$$\kappa_k^* (\tau_{nm_k}^T + \tau_{nm_k}^P + \tau_{nm_k}^C - T_{nm_k}^{\max}) = 0, 1 \leq k \leq |Q_n| \quad (29)$$

$$\nu^* \sum_{k=1}^{|Q_n|} (f_{nm_k}^* - \bar{f}_n) = 0 \quad (30)$$

根据式(23)~式(30)，可以得到 $f_{nm_k}^*$ 为

$$f_{nm_k}^* = \frac{c_{m_k} R_{nm_k} C}{T_{nm_k}^{\max} R_{nm_k} C - l_{m_k} C - s_{nm_k} R_{nm_k}}, 1 \leq k \leq |Q_n| \quad (31)$$

因此，对所有的 $n \in \mathcal{N}$ 求解上述 KKT 条件，可得计算资源分配结果 \mathbf{F}^* 。

3.5 通信资源分配子问题

将求得的计算资源分配结果 \mathbf{F}^* 代入问题式(18)，获得通信资源分配子问题为

$$\begin{aligned} \min_{\mathbf{B}} & T_3(\mathbf{B}) = T_1(\mathbf{F}^*, \mathbf{B}) \\ \text{s.t. } & \mathbf{C}1, \mathbf{C}6, \mathbf{C}7 \end{aligned} \quad (32)$$

与计算资源分配子问题相同，问题式(32)也可以被拆解为多个（不超过 N 个）相互独立的子问题。同样忽略目标函数中的常数项，这些子问题可表示为

$$\begin{aligned} \min_{B_{n1}, \dots, B_{nm_{|Q_n|}}} & \sum_{k=1}^{|Q_n|} \frac{l_{m_k}}{B_{nm_k} \text{lb}\left(1 + \frac{h_{nm_k} P_n}{\sigma_0^2}\right)}, \quad 1 \leq n \leq N \\ \text{s.t. } \widehat{\mathbf{C}}1: & \tau_{nm_k}^P + \tau_{nm_k}^T + \tau_{nm_k}^C \leq T_{nm_k}^{\max}, 1 \leq k \leq |Q_n| \\ & \widehat{\mathbf{C}}6: B_{nm_k} \geq 0, 1 \leq k \leq |Q_n| \\ & \widehat{\mathbf{C}}7: \sum_{k=1}^{|Q_n|} B_{nm_k} \leq \bar{W}_n \end{aligned} \quad (33)$$

定理 2 优化问题式(32)是凸问题。

证明 令

$$G_B(B_{n1}, \dots, B_{nm_{|Q_n|}}) = \sum_{k=1}^{|Q_n|} \frac{l_{m_k}}{B_{nm_k} \text{lb}\left(1 + \frac{h_{nm_k} P_n}{\sigma_0^2}\right)} \quad (34)$$

求二阶偏导可得

$$\frac{\partial^2 G_B}{\partial B_{ni} \partial B_{nj}} = \begin{cases} \frac{2l_i}{B_{ni}^3 \text{lb}\left(1 + \frac{h_{ni} P_n}{\sigma_0^2}\right)}, & i = j \\ 0, & i \neq j \end{cases} \quad (35)$$

因此，式(33)中目标函数的 Hessian 矩阵 $\nabla^2 G_B$ 为正定阵，由于 $\widehat{\mathbf{C}}1$ 构成凸集且 $\widehat{\mathbf{C}}6$ 和 $\widehat{\mathbf{C}}7$ 均为线性

约束条件,因此优化问题式(33)是凸问题,可利用KKT条件来求解。证毕。

优化问题式(33)的拉格朗日函数为

$$L_B(\mathbf{B}_n, \boldsymbol{\kappa}, \nu) = G_B(\mathbf{B}_n) - \sum_{k=1}^{|\mathcal{Q}_n|} \kappa_k (\tau_{nm_k}^T + \tau_{nm_k}^P + \tau_{nm_k}^C - T_{nm_k}^{\max}) - \nu \sum_{k=1}^{|\mathcal{Q}_n|} (B_{nm_k} - \bar{W}_n) \quad (36)$$

KKT条件为

$$\nabla L_B(\mathbf{B}_n^*, \boldsymbol{\kappa}^*, \nu^*) = 0 \quad (37)$$

$$\tau_{nm_k}^{T*} + \tau_{nm_k}^P + \tau_{nm_k}^C - T_{nm_k}^{\max} \leq 0, 1 \leq k \leq |\mathcal{Q}_n| \quad (38)$$

$$B_{nm_k}^* - \bar{W}_n \leq 0, 1 \leq k \leq |\mathcal{Q}_n| \quad (39)$$

$$\kappa_k^* \geq 0, 1 \leq k \leq |\mathcal{Q}_n| \quad (40)$$

$$\nu^* \geq 0 \quad (41)$$

$$B_{nm_k}^* \geq 0, 1 \leq k \leq |\mathcal{Q}_n| \quad (42)$$

$$\kappa_k^* (\tau_{nm_k}^{T*} + \tau_{nm_k}^P + \tau_{nm_k}^C - T_{nm_k}^{\max}) = 0, 1 \leq k \leq |\mathcal{Q}_n| \quad (43)$$

$$\nu^* \sum_{k=1}^{|\mathcal{Q}_n|} (B_{nm_k}^* - \bar{W}_n) = 0 \quad (44)$$

根据式(37)~式(44),可以得到 $B_{nm_k}^*$ 为

$$B_{nm_k}^* = \frac{l_{m_k} f_{nm_k} C}{(f_{nm_k} T_{nm_k}^{\max} C - s_{nm_k} f_{nm_k} - c_{m_k} C) \text{lb} \left(1 + \frac{h_{nm_k} p_{m_k}}{\sigma_0^2} \right)}, \quad 1 \leq k \leq |\mathcal{Q}_n| \quad (45)$$

对所有的 $n \in \mathcal{N}$ 求解上述KKT条件,可得带宽资源分配结果 \mathbf{B}^* ,从而可以得到原始优化问题式(14)的次优解 $\langle \mathbf{X}^*, \mathbf{F}^*, \mathbf{B}^* \rangle$ 。

3.6 算法描述与分析

综上所述,本文提出的联合卸载与资源分配优化算法如算法2所示。

算法2 联合卸载与资源分配优化算法

输入 $M, N, \langle c_m, l_m \rangle_{m \in \mathcal{M}}, \langle \bar{f}_n \rangle_{n \in \mathcal{N}}, P_m, f_m^l, h_{nm}, s_{nm},$

$\forall m \in \mathcal{M}, n \in \mathcal{N}$

输出 $\langle \mathbf{X}^*, \mathbf{F}^*, \mathbf{B}^* \rangle$

初始化 设置迭代次数 $r_2=0, r_3=0$,最大迭代次数 r_2^{\max} 和 r_3^{\max}

1) repeat

2) 执行算法1,求得 $\langle \mathbf{X}^*, \mathbf{F}^0, \mathbf{B}^0 \rangle$

3) repeat

4) 利用KKT条件求解问题式(18),即 $\mathbf{F}^* = \arg \min_{\mathbf{F}} T_2(\mathbf{F})$

5) 利用KKT条件求解问题式(32),即 $\mathbf{B}^* = \arg \min_{\mathbf{B}} T_3(\mathbf{B})$

6) 更新 $r_2=r_2+1$

7) until $r_2 \geq r_2^{\max}$

8) 更新 $r_3=r_3+1$

9) until $r_3 \geq r_3^{\max}$

10) 返回联合卸载与资源分配结果 $\langle \mathbf{X}^*, \mathbf{F}^*, \mathbf{B}^* \rangle$

以下将证明算法2的相关性质。

1) 收敛性

定理3 对于任何初始联盟分区 \mathcal{Q}_{ini} ,算法1都可以收敛获得最大化所有地面用户总效用的最终联盟分区 \mathcal{Q}^* 。

证明 每个地面用户可以根据定义2中的偏好关系,决定其想要加入的联盟并执行切换操作,这会提高系统的效用,并生成新的联盟分区。假设一次成功的切换操作可使联盟分区 \mathcal{Q}_i 转化为 \mathcal{Q}_{i+1} ,且该操作当且仅当系统效用提高时才会发生,即

$$\mathcal{Q}_i \rightarrow \mathcal{Q}_{i+1} \Leftrightarrow \mathcal{R}(\mathcal{Q}_i) < \mathcal{R}(\mathcal{Q}_{i+1}) \quad (46)$$

因此,在形成最终分区的过程中,系统效用会一直提高,即

$$\mathcal{R}(\mathcal{Q}_{ini}) < \mathcal{R}(\mathcal{Q}_1) < \mathcal{R}(\mathcal{Q}_2) < \dots < \mathcal{R}(\mathcal{Q}_i) < \mathcal{R}(\mathcal{Q}_{i+1}) < \dots < \mathcal{R}(\mathcal{Q}^*) \quad (47)$$

由于每次切换操作都会生成新分区,而地面用户和卫星的数量是有限的,从而产生的分区数是有限的。因此,算法1可保证收敛到最终联盟分区 \mathcal{Q}^* ,该分区可最大化所有地面用户的总效用。证毕。

2) 稳定性

定理4 由算法1得到的最终联盟分区 \mathcal{Q}^* 是纳什稳定分区。

证明 根据定义3,纳什稳定分区 \mathcal{Q} 表示没有任何地面用户愿意离开其当前联盟并加入 \mathcal{Q} 中的另一个联盟。假设由算法1得到的最终分区不是纳什稳定的,则必然存在地面用户 $m \in \mathcal{Q}_s$ 以及联盟 $\mathcal{Q}_k \subset \mathcal{Q}, k \neq s$,使得 $\mathcal{Q}_k \cup \{m\} >_m \mathcal{Q}_s$ 。地面用户 m 希望离开联盟 \mathcal{Q}_s 而加入联盟 \mathcal{Q}_k ,这会导致联盟分区进一步转变,与 \mathcal{Q}^* 是最终分区的事实相违背。因此,由算法1收敛后获得的联盟分区 \mathcal{Q}^* 必然是纳什稳定分区。证毕。

3) 算法复杂度

算法 1 的复杂度取决于不成功切换操作的次数和迭代次数, 而每轮迭代中不成功切换操作的次数最多为 N , 则算法 1 的计算复杂度最高为 $\mathcal{O}(Nr_1^{\max})$, 最低为 $\mathcal{O}(r_1^{\max})$ 。对于通信资源和计算资源分配子问题, 通过将全局优化问题分解为多个相互独立的局部优化子问题并行求解, 即根据式(22)~式(30)和式(36)~式(44)分别获得计算资源和通信资源分配子问题的闭式解, 从而计算复杂度仅为 $\mathcal{O}(r_2^{\max})$ 。因此, 算法 2 的总体计算复杂度为 $\mathcal{O}((r_1^{\max} + r_2^{\max})r_3^{\max})$ 到 $\mathcal{O}((Nr_1^{\max} + r_2^{\max})r_3^{\max})$ 之间。这表明 JORAO 算法复杂度较低, 而且只要迭代次数足够大, 就能收敛到局部最优解。

4 算法仿真与结果讨论

本节通过计算机仿真来评估 JORAO 算法性能, 并且将该算法与以下 4 个算法进行比较。

1) 全部本地计算 (CL, computing locally): 每个地面用户的任务均由其本身进行计算。

2) 随机卸载-资源优化分配 (RO-RAO, random offloading-resource allocation optimization) [16]: 随机选择 LEO 卫星进行卸载或本地计算, 采用优化分配方法进行通信和计算资源分配。

3) 基于 ADMM 的卸载-资源固定分配 (OADMM-FRA, offloading based on admm-fixed resource allocation) [14]: 使用文献[14]提出的 ADMM 方法进行卸载决策, 并且向地面用户固定分配通信和计算资源。

4) 基于联盟博弈的卸载-资源平均分配 (OCG-ARA, offloading based on coalition game-average resource allocation): 使用本文提出的联盟博弈方法进行卸载决策, 对于将任务卸载至 LEO 卫星的用户平均分配通信和计算资源[6]。

在计算机仿真中, 参考铱星通信系统的星座参数, 设置 LEO 卫星的轨道高度为 780 km, 轨道倾角为 86.4° , 同时服务地面用户终端的分布范围是 $1.2 \text{ km} \times 1.2 \text{ km}$ 的矩形区域。假设地面用户随机分布在该矩形区域, 且每个地面用户只有一个计算任务被卸载或者本地计算, 每个地面用户的传输功率是 23 dBm [14]。LEO 卫星使用自由空间路径损耗模型, 其最大带宽为 800 MHz [26]。考虑计算任务产生频率为 0.3 task/s, 长度范围是 [1, 6] MB, 所需的

CPU 周期范围是 [1, 6] Gcycle/s [12,14], 地面用户的计算能力是 1 Gcycle/s, LEO 卫星的计算能力是 20 Gcycle/s [12,27]。仿真参数如表 1 所示。

表 1 仿真参数

参数名称与变量	值
每个地面用户的传输功率 p_m/dBm	23
LEO 卫星的总带宽 \bar{W}_n/MHz	800
计算任务长度 l_m/MB	[1, 6]
计算任务需要的 CPU 周期数 $c_m/(\text{Gcycle} \cdot \text{s}^{-1})$	[1, 6]
地面用户的计算能力 $f_m^l/(\text{Gcycle} \cdot \text{s}^{-1})$	1
LEO 卫星的计算能力 $\bar{f}_n/(\text{Gcycle} \cdot \text{s}^{-1})$	20
LEO 卫星数目 $N/\text{个}$	[1, 9]
LEO 卫星高度 d_{nm}/km	780

首先评估 JORAO 算法的收敛性, 如图 3 所示。设置地面用户数目为 12, 计算任务所需的 CPU 周期数为 2 Gcycle/s, LEO 卫星数目分别为 3、5、7 和 9 且均在轨道中移动, 每颗卫星的初始位置沿着轨道均匀分布, 最小俯仰角为 20° [21]。

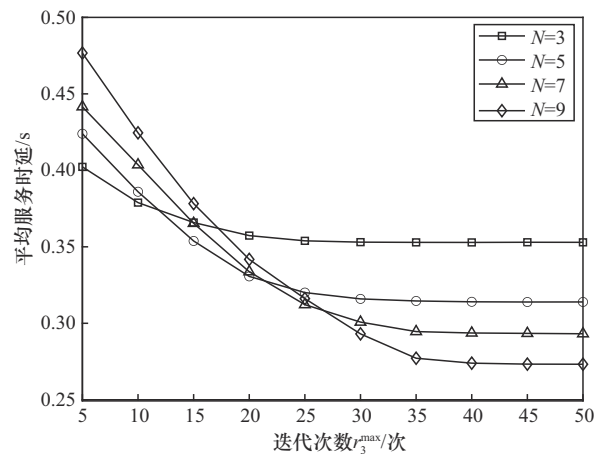


图 3 算法的收敛性

从图 3 可以看到, 当 LEO 卫星数目为 3 和 9、迭代次数 r_3^{\max} 分别为 25 和 40 时, 算法开始收敛, 说明所提算法具有良好的收敛性。此外, 在地面用户数目一定的情况下, LEO 卫星数目越多, 算法达到最终的稳定结果所需的迭代次数越多, 也就是收敛越慢。由于 JORAO 算法在每次迭代过程中没有引入新的迭代参数, 其收敛时的运行时间计算式可表示为 $T_0 = T_3 r_3^{\max} = (T_1 r_1^{\max} + T_2 r_2^{\max}) r_3^{\max}$, 其中, T_3 为 JORAO 算法的迭代周期, T_1 为算法 1 的迭代

周期, T_2 为算法 2 中资源分配子问题的迭代周期。当 LEO 卫星数目越多时, 地面用户具有更多的卸载选择, 算法 1 的 T_1 和 r_1^{\max} 增加; 对于资源分配子问题, 由于将全局优化问题分解为多个独立的局部优化子问题并行求解, 该子问题的迭代周期基本不变, 收敛所需的迭代次数 r_2^{\max} 和 r_3^{\max} 增加。因此, JORAO 算法的运行时间随着 LEO 卫星数目的增加而延长。从图 3 还可以看出, 当 LEO 卫星数目越多时, 算法最终收敛的平均服务时延越低。这是因为 LEO 卫星数目越多, 地面用户选择到能分配更多的计算资源的 LEO 卫星的概率就越大, 从而使平均服务时延降低。

当 LEO 卫星数目改变时, JORAO 算法与 OADMM-FRA 算法运行时间比较如表 2 所示。

算法	运行时间/ms			
	$N=3$	$N=5$	$N=7$	$N=9$
JORAO	62.27	78.42	94.56	112.38
OADMM-FRA	103.71	114.06	126.95	141.80

从表 2 可以看出, 随着 LEO 卫星数目增加, JORAO 算法运行时间延长。由文献[28-29]可知, 当最小俯仰角为 20° 时, 铱星的平均覆盖时间约为 400 ms, 而 JORAO 算法的收敛时间远小于 400 ms。因此, JORAO 算法在迭代稳定之前, LEO 卫星的覆盖条件不会发生变化。从表 2 还可以看出, OADMM-FRA 算法运行时间始终大于 JORAO 算法。由文献[14]可知, OADMM-FRA 算法使用 ADMM 法获得次优的卸载决策, 复杂度为 $O((M^3 + MN + 2M)r_3^{\max})$, 高于 JORAO 算法复杂度, 因此, OADMM-FRA 算法迭代稳定的运行时间也较长。本文考虑的 LEO 卫星边缘计算场景中使用 SDN 实现全局控制, 执行 JORAO 算法的 SDN 服务器通常可部署于 LEO 卫星; 从算法 1 和算法 2 可知, 执行算法所需的 LEO 卫星与地面用户之间的信息交互量与地面用户的计算任务量相比可以忽略不计。因此, 本文所提算法的控制维护代价较小。

不同算法的平均服务时延随地面用户数目的变化如图 4 所示, 其中, LEO 卫星数目为 5, 计算任务所需的 CPU 周期数为 2 Gcycle/s。

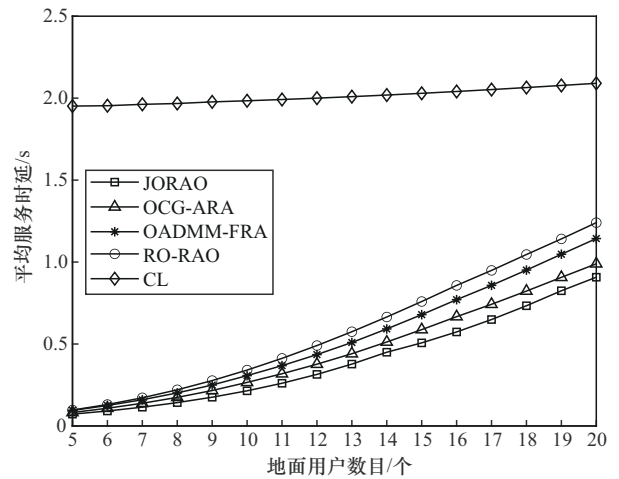


图 4 不同算法的平均服务时延随地面用户数目的变化

从图 4 中可以看出, 随着地面用户数目的增加, 5 个算法的平均服务时延均增加。这是因为, 增加地面用户数目意味着增加计算任务数, 而每个计算任务被分配的通信和计算资源有限, 从而平均服务时延增加。此外, 随着地面用户数目的进一步增加, 其可能超过 LEO 卫星的最大计算能力, 此时部分地面用户将选择本地处理计算任务, 这会导致平均服务时延的进一步增加。比较 5 个算法可以看出, 随着地面用户数目的增加, JORAO 算法始终具有最低的平均服务时延, 分别比 RO-RAO、OADMM-FRA 和 OCG-ARA 平均低 32.52%、25.95% 和 14.44%。

不同算法的平均服务时延随 LEO 卫星数目的变化如图 5 所示, 其中, 地面用户数目为 12, 计算任务所需的 CPU 周期数为 2 Gcycle/s。

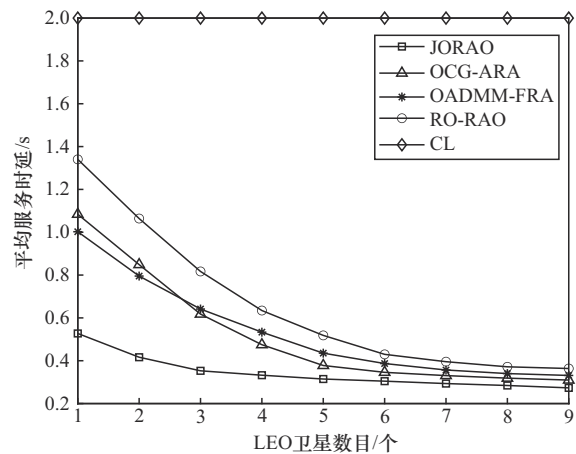


图 5 不同算法的平均服务时延随 LEO 卫星数目的变化

从图 5 中可以看出, 除 CL 外, 另外 4 个算法随着 LEO 卫星数目的增加, 平均服务时延降低。这

是因为LEO卫星数目越多，地面用户有更多的机会选择合适的LEO卫星进行计算任务卸载。从图5还可以看出，当LEO卫星数目改变时，JORAO算法的平均服务时延始终低于其他算法，这是因为JORAO算法除了基于联盟博弈方法进行卸载决策外，还向将计算任务卸载至LEO卫星的地面用户优化分配通信和计算资源，降低了平均服务时延。随着LEO卫星数目的进一步增加，JORAO与OCG-ARA、OADMM-FRA和RO-RAO之间的性能差距变小，这是因为地面用户总数保持不变，计算任务所需的资源量不变，而随着LEO卫星数目的增加，选择进行任务卸载的地面用户都能够获得充足的资源，平均服务时延降低的趋势逐渐放缓。特别地，当LEO卫星数目为6时，JORAO的平均服务时延分别比RO-RAO、OADMM-FRA和OCG-ARA低29.15%、21.29%和19.37%。

不同算法的平均服务时延随计算任务长度的变化如图6所示，其中，地面用户数目为12，LEO卫星数目为5。

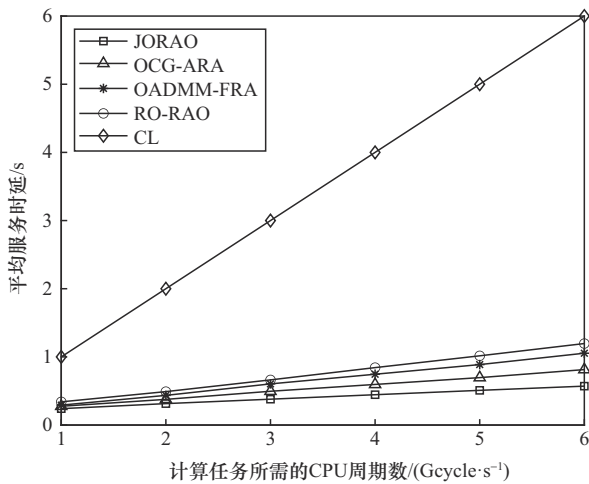


图6 不同算法的平均服务时延随计算任务长度的变化

从图6可以看出，随着计算任务所需的CPU周期数增加，5个算法的平均服务时延也增加。计算任务所需的CPU周期数越大，导致网络中的计算负载越高，从而增加了计算时延。在5个算法中，CL具有最高的平均服务时延，这是因为CL只能在本地处理计算任务，由式(12)可知，当计算任务所需的CPU周期数增加时，计算时延增加。在RO-RAO和OCG-ARA中，地面用户的计算任务可以在本地计算或者卸载到LEO卫星处理。在OADMM-

FRA中，地面用户的计算任务可以在本地计算，也可以卸载至LEO卫星处理，还可以卸载至中心云服务器处理。随着地面用户的计算需求逐渐增大，其可能超过LEO卫星的计算能力。在RO-RAO中，由于随机选择LEO卫星进行卸载或本地计算，地面用户的计算任务不能获得充足的资源。在OCG-ARA中，被卸载的计算任务平均分配计算资源。在OADMM-FRA中，固定向地面用户分配计算资源。JORAO则能够根据计算任务需求及时地优化计算资源的分配。因此，JORAO的平均服务时延低于RO-RAO、OADMM-FRA和OCG-ARA的平均服务时延。特别地，当计算任务所需的CPU周期数为2 Gcycle/s时，JORAO的平均服务时延分别比RO-RAO、OADMM-FRA和OCG-ARA低35.87%、27.74%和16.51%。由文献[14]可知，在OADMM-FRA算法中，地面用户可将计算任务卸载至LEO卫星处理，若由于LEO卫星分配的资源有限而使得向LEO卫星卸载任务的服务时延超过LEO卫星覆盖时间，此时可将计算任务卸载至服务时延低于LEO卫星覆盖时间的中心云服务器。然而，由于OADMM-FRA中的LEO卫星和中心云服务器均向地面用户固定分配通信和计算资源，因此其平均服务时延高于OCG-ARA和JORAO算法。

不同算法的任务卸载成功率和平均服务时延随俯仰角的变化分别如图7和图8所示，其中，地面用户数目为12，LEO卫星数目为5。根据式(2)~式(4)，LEO卫星与地面用户的最长通信时间（即LEO卫星的覆盖时间）与它们之间的俯仰角相关，俯仰角越大，LEO卫星与地面用户之间的最长通信时间越长^[14,29]。

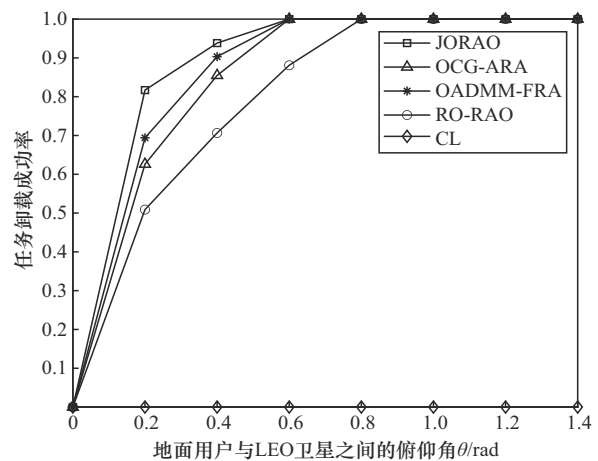


图7 不同算法的任务卸载成功率随俯仰角的变化

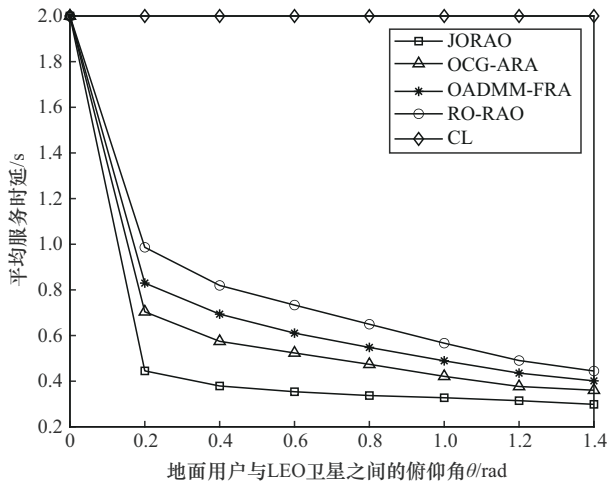


图8 不同算法的平均服务时延随俯仰角的变化

从图7可以看到,当 $\theta=0$ 时,RO-RAO、OADMM-FRA、OCG-ARA和JORAO算法的任务卸载成功率为0,并且其任务卸载成功率随着俯仰角的增加而增大。从图8可以看到,RO-RAO、OADMM-FRA、OCG-ARA和JORAO算法的平均服务时延随着俯仰角的增加而降低,当 $\theta=1.4$ rad(即 80.21°)时,地面用户的平均服务时延达到最低。这是因为,当俯仰角较小时,LEO卫星与地面用户之间的最长通信时间也缩短,这表明只能有部分地面用户能够将计算任务成功卸载至LEO卫星或通过LEO卫星卸载至中心云服务器,剩余地面用户的计算任务或者卸载失败或者只能在本地计算,导致平均服务时延增加。特别地,当 $\theta=0$ 时,地面用户与LEO卫星之间的通信时间不能支持其执行任务卸载,所以地面用户的计算任务只能在本地执行,因此任务卸载成功率为0。随着 θ 的增加,LEO卫星与地面用户之间的最长通信时间延长,地面用户执行计算卸载的成功率增加,平均服务时延降低。与其他4个算法相比,JORAO算法有着最高的平均任务卸载成功率和最低的平均服务时延。

5 结束语

本文针对一种LEO卫星边缘计算场景,提出了联合卸载与资源分配优化算法,考虑了LEO卫星的有限覆盖时间和通信计算能力,以最小化所有地面用户的平均服务时延为目标,联合优化卸载策略、通信和计算资源分配。将任务卸载与资源分配的联合优化问题分解为卸载决策和资源分配子问题,使用交替优化方法,获得原始优化问题的次优

解。对于任务卸载决策子问题,将其建模为联盟博弈模型,当博弈达到纳什均衡时,获得最小化平均服务时延的所有地面用户的卸载策略;对于资源优化子问题,使用拉格朗日乘子法获得通信和计算资源分配结果。仿真结果表明,当改变地面用户数目、LEO卫星数目、计算任务长度以及俯仰角时,JORAO算法与已有算法相比均能显著降低地面用户的平均服务时延、提高任务卸载成功率。特别地,当改变地面用户数目时,JORAO算法的平均服务时延分别比RO-RAO、OADMM-FRA和OCG-ARA算法低32.52%、25.95%和14.44%。

参考文献:

- [1] LIU J Y, ZHAO X W, QIN P, et al. Joint dynamic task offloading and resource scheduling for WPT enabled space-air-ground power Internet of Things[J]. IEEE Transactions on Network Science and Engineering, 2022, 9(2): 660-677.
- [2] AZZARELLI T. OneWeb global access[C]//Proceedings of the IEEE Global Conference on Space and the Society. Piscataway: IEEE Press, 2016: 1-24.
- [3] FOREMAN V L, SIDDIQIA, WECK O D. Large satellite constellation orbital debris impacts: case studies of OneWeb and SpaceX proposals[C]//Proceedings of the AIAA SPACE and Astronautics Forum and Exposition. Reston: AIAA, 2017: 5200.
- [4] WOOD L, LOU Y X, OLUSOLA O. Revisiting elliptical satellite orbits to enhance the O3b constellation[J]. JBIS - Journal of the British Interplanetary Society, 2014, 67(3): 110-118.
- [5] 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.
- [6] ZHANG Z J, ZHANG W Y, TSENG F H. Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques[J]. IEEE Network, 2019, 33(1): 70-76.
- [7] CAO X L, YANG B, SHEN Y L, et al. Edge-assisted multi-layer offloading optimization of LEO satellite-terrestrial integrated networks[J]. IEEE Journal on Selected Areas in Communications, 2023, 41(2): 381-398.
- [8] LAKEW D S, TRAN A T, DAO N N, et al. Intelligent self-optimization for task offloading in LEO-MEC-assisted energy-harvesting-UAV systems[J]. IEEE Transactions on Network Science and Engineering, 2024, PP(99): 1-14.
- [9] HAO Y Y, SONG Z Y, ZHENG Z, et al. Joint communication, computing, and caching resource allocation in LEO satellite MEC networks[J]. IEEE Access, 2023, 11: 6708-6716.
- [10] TUN Y K, KIM K T, ZOU L Y, et al. Collaborative computing services at ground, air, and space: an optimization approach[J]. IEEE Transactions on Vehicular Technology, 2024, 73(1): 1491-1496.
- [11] LI C C, ZHANG Y S, HAO X K, et al. Jointly optimized request dispatching and service placement for MEC in LEO network[J]. China Communications, 2020, 17(8): 199-208.

- [12] WANG Y X, YANG J, GUO X Y, et al. A game-theoretic approach to computation offloading in satellite edge computing[J]. IEEE Access, 2020, 8: 12510-12520.
- [13] LIN X, LIU A J, HAN C, et al. LEO satellite and UAVs assisted mobile edge computing for tactical ad-hoc network: a game theory approach[J]. IEEE Internet of Things Journal, 2023, 10(23): 20560-20573.
- [14] TANG Q Q, FEI Z S, LI B, et al. Computation offloading in LEO satellite networks with hybrid cloud and edge computing[J]. IEEE Internet of Things Journal, 2021, 8(11): 9164-9176.
- [15] SONG Z Y, HAO Y Y, LIU Y W, et al. Energy-efficient multiaccess edge computing for terrestrial-satellite Internet of things[J]. IEEE Internet of Things Journal, 2021, 8(18): 14202-14218.
- [16] JIA M, ZHANG L, WU J, et al. Joint computing and communication resource allocation for edge computing towards Huge LEO networks[J]. China Communications, 2022, 19(8): 73-84.
- [17] LI P X, WANG Y C, WANG Z N. A game-based joint task offloading and computation resource allocation strategy for hybrid edge-cloud and cloudy-edge enabled LEO satellite networks[C]//Proceedings of the 2022 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE Press, 2022: 868-873.
- [18] XIE R C, TANG Q Q, WANG Q N, et al. Satellite-terrestrial integrated edge computing networks: architecture, challenges, and open issues[J]. IEEE Network, 2020, 34(3): 224-231.
- [19] WU J, JIA M, GUO Q, et al. Efficient resource management based on DQN in LEO satellite edge computing system[C]//Proceedings of the 2023 IEEE Globecom Workshops (GC Wkshps). Piscataway: IEEE Press, 2023: 135-140.
- [20] SHI Y P, CAO Y R, LIU J J, et al. A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks[J]. IEEE Network, 2019, 33(1): 29-35.
- [21] LYU Y F, LIU Z, FAN R F, et al. Optimal computation offloading in collaborative LEO-IoT enabled MEC: a multiagent deep reinforcement learning approach[J]. IEEE Transactions on Green Communications and Networking, 2023, 7(2): 996-1011.
- [22] ABDI A, LAU W C, ALOUINI M S, et al. A new simple model for land mobile satellite channels: first- and second-order statistics[J]. IEEE Transactions on Wireless Communications, 2003, 2(3): 519-528.
- [23] STOJANOVIC M, CHAN V. Adaptive power and rate control for satellite communications in Ka band[C]//Proceedings of the 2002 IEEE International Conference on Communications. Piscataway: IEEE Press, 2002: 2967-2972.
- [24] YOU L, LI K X, WANG J H, et al. Massive MIMO transmission for LEO satellite communications[J]. IEEE Journal on Selected Areas in Communications, 2020, 38(8): 1851-1865.
- [25] OSBORNE M J, RUBINSTEIN A. A course in game theory[M]. Cambridge: MIT Press, 1994.
- [26] DENG R Q, DI B Y, CHEN S Z, et al. Ultra-dense LEO satellite offloading for terrestrial networks: how much to pay the satellite operator?[J]. IEEE Transactions on Wireless Communications, 2020, 19(10): 6240-6254.
- [27] 3GPP TR 38.811. Study on new radio (NR) to support non terrestrial networks (Release15) (V0.3.0)[S]. 2017.
- [28] 杨斌, 何锋, 靳瑾, 等. LEO卫星通信系统覆盖时间和切换次数分析[J]. 电子与信息学报, 2014, 36(4): 804-809.
- YANG B, HE F, JIN J, et al. Analysis of coverage time and handoff number on LEO satellite communication systems[J]. Journal of Electronics & Information Technology, 2014, 36(4): 804-809.
- [29] SEYEDI Y, SAFAVI S M. On the analysis of random coverage time in mobile LEO satellite communications[J]. IEEE Communications Letters, 2012, 16(5): 612-615.

[作者简介]



夏玮玮 (1975-), 女, 江苏句容人, 博士, 东南大学副研究员、博士生导师, 主要研究方向为无线网络资源管理、边缘计算、泛在网络与短距离无线通信等。



胡静 (1975-), 女, 江苏扬州人, 博士, 东南大学副研究员、硕士生导师, 主要研究方向为泛在网络与短距离无线通信、物联网等。



宋铁成 (1967-), 男, 江苏张家港人, 博士, 东南大学教授、博士生导师, 主要研究方向为车联网、无线网络资源优化等。