

LEO 星座边缘计算网络中的动态计算卸载策略

高玉芳¹, 姬智², 赵康健¹, 李文峰¹, 胡佩聪³

(1. 南京大学电子科学与工程学院, 江苏 南京 210046; 2. 陆军工程大学通信工程学院, 江苏 南京 210007;
3. 中国人民解放军 63893 部队, 河南 洛阳 471003)

摘要: 在卫星边缘计算网络中, 当过多用户通过同一信道接入卫星时, 产生的同信道干扰会导致边缘计算性能下降。为了解决该难题, 在动态环境低地球轨道 (LEO) 星座边缘计算网络的系统模型下, 提出了一种基于随机博弈的多用户计算卸载策略。在考虑用户的自私性、星地信道的随机特性和地面用户接入的动态特性的前提下, 从博弈论的角度, 将动态环境下地面用户的卸载决策过程表述为随机博弈, 证明了所制定的随机博弈等价于具有至少一个纳什均衡 (NE) 的加权势博弈, 并且 NE 最小化系统开销。为了在动态环境下以分布式方式达到 NE, 基于随机学习设计了一种智能随机学习算法, 以高效达到所提随机博弈的 NE。仿真结果表明, 与基准算法相比, 所提算法能够显著降低同信道干扰和系统开销, 并达到接近最优的性能。

关键词: 空间物联网; 边缘计算; 计算卸载; 动态环境; 势博弈

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024065

Dynamic computing offloading strategy in LEO constellation edge computing network

GAO Yufang¹, JI Zhi², ZHAO Kanglian¹, LI Wenfeng¹, HU Peicong³

1. School of Electronic Science and Engineering, Nanjing University, Nanjing 210046, China
2. College of Communications Engineering, Army Engineering University of PLA, Nanjing 210007, China
3. PLA 63893 Troops, Luoyang 471003, China

Abstract: In satellite edge computing networks, when too many ground users access the satellite through the same channel, the resulting co-channel interference will lead to edge computing performance degradation. To address this problem, a multi-user computing offloading strategy based on stochastic game was proposed under the system model of dynamic environment low earth orbit constellation edge computing network. On the premise of considering the selfishness of users, the stochastic characteristics of the satellite-ground channel and the dynamic nature of ground user access, from the perspective of game theory, the offloading decision-making process of ground users in the dynamic environment was formulated as a stochastic game. It was proved that the formulated stochastic game was equivalent to a weighted potential game with at least one Nash Equilibrium (NE), and the NE minimized the system overhead. In order to achieve NE in a distributed manner under dynamic environment, an intelligent stochastic learning algorithm based on the stochastic learning was designed to efficiently achieve NE for the proposed stochastic game. Simulation results show that compared to the benchmark algorithm, the proposed algorithm can significantly reduce the co-channel interference and the system overhead, and achieve near-optimal performance.

Keywords: Internet of space things, edge computing, computing offloading, dynamic environment, potential game

收稿日期: 2023-10-12; 修回日期: 2024-01-14

通信作者: 赵康健, zhaokanglian@nju.edu.cn

基金项目: 国家自然科学基金资助项目 (No.62131012); 中央高校基本科研业务费专项资金资助项目 (No.021014380187)

Foundation Items: The National Natural Science Foundation of China (No.62131012), The Fundamental Research Funds for the Central Universities (No.021014380187)

0 引言

随着 6G 通信的快速发展, 低地球轨道 (LEO, low earth orbit) 星座网络引起了人们的广泛关注^[1-2]。由于卫星可以实现全球覆盖的特点, 非地面网络 (NTN, non-terrestrial network) 系统^[3]被设想为地面信息系统的有力补充, 为偏远、农村或过载地区 (尤其是现有基础设施无法覆盖、经济或技术上不可行的地区) 提供了无处不在的全球连接服务。通过扩展提供异构连接到空间的物联网 (IoT, Internet of things) 概念, 提出了一种名为空间物联网 (IoST, Internet of space things)^[4]的新范式。IoST 作为 NTN 支持的主要应用程序之一, 可以提供监视、侦察、太空回程和信息物理集成等服务。

移动边缘计算 (MEC, mobile edge computing) 将计算和存储资源转移到更靠近用户的位置, 已经为物联网做出了很多贡献^[5-7]。近年来, 在空天地一体化网络无处不在的连接和全球覆盖的驱动下, 边缘计算发生了从地面边缘计算向 IoST 边缘计算的转变^[8-11], 可提供无处不在、宽频和可靠的计算服务。文献^[9-10]中, LEO 卫星仅充当透明中继节点, 没有充分利用卫星潜在的计算优势。文献^[11]从功耗、抗辐射、成本等几个方面深入讨论了在太空放置边缘服务器的可行性。文献^[1]研究了 Mega-LEO 星座中的计算范式, 其中, 地面/非地面节点可以通过将计算任务卸载到 LEO 星座的轨道边缘平台而受益。然而, 这也给通信过程带来了更多挑战。当大量地面用户通过同一共享信道上其计算任务时, 在轨计算^[11]的优势就会被削弱。严重的信道干扰会降低任务上传速率并增加地面用户的能耗。因此, 当地面用户争夺有限数量的星地信道进行边缘计算时, 设计高效的计算卸载策略就变得至关重要。

现有的许多工作研究了 LEO 星座网络的计算卸载。例如, 文献^[12]通过凸优化的方法实现了计算卸载过程中用户能耗的最小化。文献^[13]研究了具有地面边缘计算和卫星边缘计算能力的双边计算星地网络, 通过以最低的成本将任务分配给边缘服务器来优化能耗和时延。文献^[14]利用凸优化和深度强化学习解决了带宽分配和卸载位置的优化问题。文献^[15]提出了一种具有 3 层计算架构的混合云和边缘计算 LEO 星座网络, 并通过凸优化方法

优化卸载决策, 以最小化系统总能耗。

上述研究都假设信道是时不变的, 并且服务的用户是确定不变的。然而, 在真实的网络环境中, 这样的假设是不现实的。星地链路中的云、雾、湍流等固有的自然特性使星地信道随着时间的推移而不断变化。同时, 复杂的网络环境决定了用户接入和离开系统是随机的, 即用户的状态也是动态的。此外, 需要进一步开发新的分布式方法。因为随着在轨卫星数量的不断增加, 大规模计算对集中式方法来说是灾难性的。而且, 所有地面用户都被划分为不同的主体, 具有自私倾向, 集中式方法不适合动态环境中的快速决策。为了捕捉网络环境的动态性, 本文运用博弈论算法来分析动态条件下用户的计算卸载决策过程。

博弈论^[16]是一种被广泛应用的分布式资源管理方法, 用于建模和分析一组目标相互冲突的理性决策者之间的复杂决策过程。值得注意的是, 与深度强化学习等分布式方法相比, 博弈论不需要极高的计算资源, 而且在处理不完全信息事件方面具有显著的优势。在运用博弈论解决星地计算卸载方面, 文献^[17-18]分别采用完全信息静态博弈和精确势博弈实现了 LEO 卫星边缘计算系统中的计算卸载。文献^[19-20]通过调用斯坦伯格 (Stackelberg) 博弈分别设计了面向超密集 LEO 星座网络和 Starlink 卫星系统的星地数据卸载。但上述工作仍然忽略了实际环境的动态性和用户访问的不确定性等特点。

基于上述讨论, 针对环境的动态性、用户访问的不确定性和由前两者引起的信息不完全性问题, 本文设计了分布式动态随机算法。该算法考虑了在实际中随时间变化的信道状态、活跃用户访问, 以及不完整信息。本文主要研究工作如下。

1) 针对动态环境下 LEO 星座边缘计算网络, 在环境高动态性和活跃用户访问约束条件下, 通过设计高效的计算卸载策略, 建立了最小化网络总开销的系统模型。

2) 从博弈论的角度, 将系统开销最小化问题建模为一个随机博弈。为了证明纳什均衡 (NE, Nash equilibrium) 在形成的随机博弈中的存在性, 证明了其与具有至少一个 NE 的加权势博弈的等价性, 并证明了 NE 可以最小化系统开销。用户在动态环境下以分布式方式达到 NE 是一个具有挑战性

的问题, 现有博弈论算法可能永远无法达到。为了找到NE, 本文提出了一种用于动态环境的智能随机学习算法, 以完全分布式的方式进行了计算卸载, 并证明了所提算法的复杂度。

3) 通过仿真, 评估了本文所提算法的性能。通过分析信道选择的收敛过程, 证明了算法的收敛性。通过计算不同用户和信道下的计算开销, 证明了与基准算法相比, 所提算法可实现最小的网络总开销。

1 系统模型与问题表述

1.1 系统模型

在陆地通信尚未覆盖的地区, 例如孤岛、偏远山区或遭受自然灾害的地方等, 陆地计算资源的稀缺可能无法满足所有用户的通信需求, 因此, 需要进行计算卸载。LEO星座边缘计算网络系统模型如图1所示, 卫星搭载边缘服务器以提供计算服务, 地面用户通过将任务卸载到卫星以实现边缘计算。考虑卫星与地面用户之间采用直接接入的互操作模式^[21], 数据传输采用正交频分多址^[22]。将地面用户集合表示为 $\mathcal{N} = \{1, 2, \dots, N\}$; 可用信道数为 M , 信道集合可以表示为 $\mathcal{M} = \{1, 2, \dots, M\}$ 。用户有2种选择: 本地计算或在轨计算。将卸载策略表示为 $s_n \in \{0\} \cup M$, 其中, $s_n = 0$ 表示用户 n 在本地计算任务, $s_n > 0$ 表示用户 n 通过信道 s_n 进行计算卸载, 任务在卫星上计算。系统参数如表1所示。

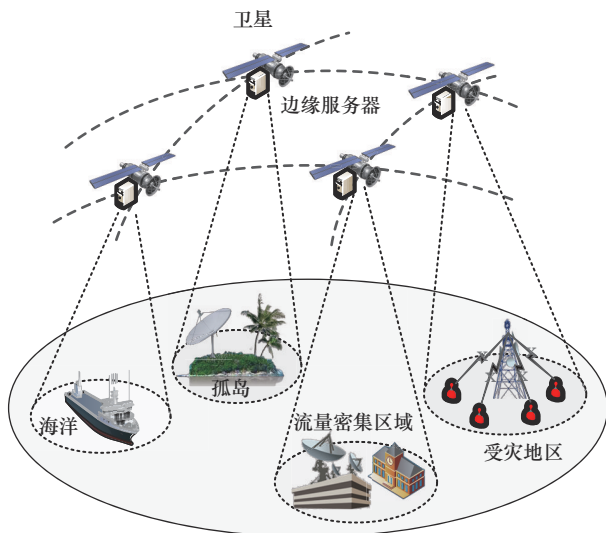


图1 LEO星座边缘计算网络系统模型

表1	系统参数	
参数	含义	
N	地面用户数量	
M	信道数量	
α	路径损耗指数	
p_n	用户 n 的传输功率	
$g_{n,\text{sat}}$	地面用户 n 与卫星间的信道增益	
φ_n	用户 n 的活跃概率	
$d_{n,\text{sat}}$	用户 n 和卫星之间的距离	
\mathcal{A}	活跃用户集合	
\bar{u}_n	随机博弈 \mathcal{G} 的效用函数	
$s_{\mathcal{A}}$	所有活跃用户的策略组合	
s	所有用户的策略组合	
$\Phi(s_n, s_{-n})$	博弈 \mathcal{G} 的势函数	
v_n	用户 n 的策略选择概率向量	
b	学习步长	
Ξ	样本空间中的事件	

假设星地信道遵循莱斯衰落^[23]。星地信道的信道功率增益为 $g_{n,\text{sat}}[t] = (d_{n,\text{sat}}[t])^{-\alpha} \beta_{n,\text{sat}}$, 其中, $d_{n,\text{sat}}[t] = \sqrt{r_c^2 + (r_c + h)^2 - 2r_c(r_c + h)\cos\gamma[t]}$ 是用户 n 到卫星的距离, r_c 是地球半径, h 是用户与LEO卫星轨道之间的距离, γ 是地心角, $\gamma[t] = \arccos\left(\frac{r_c}{r_c + h}\cos\theta[t]\right) - \theta[t]$, $\theta[t]$ 是用户与卫星之间的仰角; α 是路径损耗指数; $\beta_{n,\text{sat}}$ 是莱斯衰落因子。由于环境的动态性, 瞬时随机系数 $\beta_{n,\text{sat}}$ 在任何时隙都会发生变化。考虑一个更通用的模型, 用户在有任务时处于活跃状态, 没有任务时处于不活跃状态。假设用户的活跃状态由开关模型描述, 设置活跃用户集合为 $\mathcal{A} = \{n \in \mathcal{N}: a_n[t] = 1\}$, 用户 n 活跃的概率为 φ_n , 则用户 n 不活跃的概率为 $1 - \varphi_n$ 。

定义一个概率空间 $(\Omega, \mathcal{I}, \mathbb{P})$, 其中, Ω 是所有状态的样本空间, \mathcal{I} 是集合 Ω 上的最小 σ 代数, \mathbb{P} 是 (Ω, \mathcal{I}) 的概率度量。考虑样本空间 Ω 中的一个事件, 记为 Ξ 。 $\Theta(\Xi) = [\mathbf{a}(\Xi), \mathbf{g}(\Xi)]: \Omega \rightarrow 2^N \times \mathbb{R}^N$ 是随机变量, 其中, $\mathbf{a} = [a_n]_{n \in \mathcal{N}}$, a_n 表示用户 n 的活跃状态, 服从参数 φ_n 的开关分布; $\mathbf{g} =$

$\{g_{n,\text{sat}}\}_{\forall n \in \mathcal{A}}$ 服从莱斯分布。假设用户 $n \in \mathcal{A}$ 通过信道 $s_n > 0$ 将其计算任务上传到卫星进行在轨计算, 则所有活跃用户的策略组合为 $s_{\mathcal{A}} = [s_n]_{\forall n \in \mathcal{A}}$ 。在时隙 t , 事件 $\Xi[t] \in \Omega$ 可实现的吞吐量为

$$R_n(s_{\mathcal{A}}, \Xi[t]) = B \ln \left(1 + \frac{p_n g_{n,\text{sat}}[t]}{\sum_{j \in \mathcal{A} \setminus \{n\} : s_j = s_n} p_j g_{j,\text{sat}}[t] + \sigma_0} \right) \quad (1)$$

其中, B 是带宽, p_n 是用户 n 的发射功率, $g_{n,\text{sat}}[t]$ 是用户 n 到卫星的信道增益, σ_0 是背景噪声功率,

$\sum_{j \in \mathcal{A} \setminus \{n\} : s_j = s_n} p_j g_{j,\text{sat}}[t]$ 是多个用户在同一信道上卸载

形成的相互干扰。从式(1)可以看出, 用户之间的选择策略是耦合的。如果过多的用户选择同一个信道进行计算卸载, 同信道干扰很大, 会导致较低的上传速率。

1.2 计算模型

每个用户 $n \in \mathcal{A}$ 都有一个要执行的计算任务 $\Gamma_n = (W_n, D_n^{\text{loc}}, D_n^{\text{sat}})$, 其中, W_n 表示任务数据量的大小, D_n^{loc} 表示本地计算能力或完成任务所需要的 CPU 周期数, D_n^{sat} 表示卫星的计算能力。用户任务可以在本地执行, 也可以发送到卫星进行在轨计算。下面分别对 2 种方式的成本进行计算。

将任务卸载到卫星上有 3 个过程, 即传输过程、计算过程和传播过程。成本包括时延和能耗两部分。在时隙 t , W_n 所需的传输时延和能耗分别为 $T_{n,1}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) = \frac{W_n}{R_n(s_{\mathcal{A}}, \Xi[t])}$ 和

$$E_{n,1}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) = \frac{p_n W_n}{R_n(s_{\mathcal{A}}, \Xi[t])};$$
 在轨计算所需时延

为 $T_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) = \frac{D_n^{\text{sat}}}{f_n^{\text{sat}}}$, 其中, f_n^{sat} 为卫星分配给

用户 n 的计算能力, 由于处理器的能耗与 CPU 频率的平方成正比^[15], 因此任务执行的能耗可以表示为 $E_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) = \eta_n D_n^{\text{sat}}$, 其中, η_n ^[17] 是由在轨边缘服务器的 CPU 硬件架构决定的常数; 传播时延可以表示为 $T_{n,3}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) = \frac{d_{n,\text{sat}}[t]}{c}$, 其中, c 是光速。

考虑到时延和能耗, 在时隙 t , 卫星上用户 n 的在轨计算总成本为

$$\begin{aligned} V_n^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) = & \mu_n^E [E_{n,1}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) + E_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t])] + \\ & \mu_n^T [T_{n,1}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) + T_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) + T_{n,3}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t])] = \\ & \frac{(\mu_n^T + \mu_n^E p_n) W_n}{R_n(s_{\mathcal{A}}, \Xi[t])} + \mu_n^E E_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) + \\ & \mu_n^T [T_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) + T_{n,3}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t])] \quad (2) \end{aligned}$$

其中, $\mu_n^T, \mu_n^E \in (0, 1)$ 且 $\mu_n^T + \mu_n^E = 1$, μ_n^T 和 μ_n^E 表示时延和能耗对用户 n 策略选择过程的相对权重。如果用户对时延要求较高但不关心能耗 (例如处理视频流、实时数据等), 则 μ_n^T 的比例较高; 如果用户对能耗要求较高但对时延不敏感或者用户处于低电量状态, 则 μ_n^E 的比例较高。

假设 f_n^{loc} 为用户 n 的本地计算能力, 则本地计算的时延为 $T_n^{\text{loc}} = \frac{D_n^{\text{loc}}}{f_n^{\text{loc}}}$; 本地计算的能耗可以表示为 $E_n^{\text{loc}} = \eta_n D_n^{\text{loc}}$, 故本地计算总成本可以表示为

$$V_n^{\text{loc}} = \mu_n^T T_n^{\text{loc}} + \mu_n^E E_n^{\text{loc}} \quad (3)$$

根据式(1)和上面讨论的计算模型, 如果过多用户选择相同的信道接入策略进行计算卸载, 其传输速率会快速下降, 同时会增加用户的成本。此时, 如果部分用户选择本地计算则为最佳选择。由于用户访问的随机性和信道的时变性, 很难准确把握其他用户的工作状态, 并且各个用户之间相互依赖、相互影响, 使问题变得更具挑战性, 因此本文采用博弈论来描述和分析每个用户的计算策略, 运用博弈论算法为动态环境下的 LEO 星座网络设计相应的用户计算卸载策略。本文的目标是通过优化动态环境下的卸载策略以最小化网络总开销。所提出的资源分配问题可以表示为

$$\begin{aligned} \min \{ & V_n^{\text{sat}}, V_n^{\text{loc}} \} \\ \text{s.t. } & 0 \leq t \leq T \\ & 0 \leq \varphi \leq 1 \\ & 1 \leq n \leq N \\ & 1 \leq m \leq M \\ & \mathcal{A} \in \mathcal{N} \quad (4) \end{aligned}$$

其中, 卸载策略和信道选择都与时隙 t 的变化有关, 因此使用传统的凸优化技术很难解决这一问题。具体而言, 所提出的问题具有以下特点: 1) 地面用户的工作状态是动态变化的; 2) 用户倾向于选择条件较好的信道上传任务并进行在轨计算, 但由于

信道的时变性和用户状态的动态性,这一问题变得复杂;3)信息不完全性(每个用户不可能知道其他用户的工作状态和其他用户选择的信道情况)。因此,直接解决所提问题是相当棘手的。

2 博弈模型及NE分析

2.1 博弈模型

每个活跃用户 $n \in \mathcal{A}$ 根据其他用户的策略独立调整其卸载策略以最小化成本。特别地,给定概率空间 $(\Omega, \mathcal{I}, \mathbb{P})$ 中的事件 Ξ , 可以定义用户 n 在时隙 t 的代价函数为

$$u_n^0(s_n, s_{\mathcal{A}\setminus\{n\}}, \Xi[t]) = \begin{cases} V_n^{\text{loc}}, & s_n = 0 \\ V_n^{\text{sat}}(s_n, s_{\mathcal{A}\setminus\{n\}}, \Xi[t]), & s_n > 0 \end{cases} \quad (5)$$

其中, s_n 表示用户 n 的卸载策略, $s_{\mathcal{A}\setminus\{n\}}$ 表示除用户 n 之外的其他所有活跃用户的策略组合。由式(1)~式(3)可知, 用户 n 在信道上受到的干扰(即 $J_n(s_{\mathcal{A}}, \Xi[t]) = \sum_{j \in \mathcal{A}\setminus\{n\}; s_j = s_n} p_n g_{n,\text{sat}}[t]$) 是影响用户是否卸载的关键因素。如果信道干扰低于特定阈值, 则用户更倾向于在轨计算。具体来说, 基于类似文献[16]的证明方法, 可以得出下述结论。

给定计算卸载策略组合 $s_{\mathcal{A}} = [s_n]_{\forall n \in \mathcal{A}}$, 如果信道上的干扰 $J_n(s_{\mathcal{A}}, \Xi[t]) = \sum_{j \in \mathcal{A}\setminus\{n\}; s_j = s_n} p_n g_{n,\text{sat}}[t]$ 满足 $J_n(s_{\mathcal{A}}, \Xi[t]) \leq A_n[t]$, 则用户 n 倾向于选择在轨计算。其中, $A_n[t] = \frac{p_n g_{n,\text{sat}}[t]}{2^{\psi_n} - 1} - \sigma_0$, $\psi_n = \frac{(\mu_n^E p_n + \mu_n^T) W_n}{B(\mu_n^T(T_n^{\text{loc}} - T_{n,2}^{\text{sat}} - T_{n,3}^{\text{sat}}) + \mu_n^E(E_n^{\text{loc}} - E_{n,2}^{\text{sat}}))}$ 。具体证明如下。

当用户 n 的在轨计算成本不超过其本地计算成本(即 $V_n^{\text{sat}} \leq V_n^{\text{loc}}$) 时, 在轨计算对用户 n 更有利, 因此用户 n 更倾向于计算卸载。根据式(2)和式(3)可得

$$\frac{(\mu_n^T + \mu_n^E p_n) W_n}{R_n(s_{\mathcal{A}}, \Xi[t])} + \mu_n^T (T_{n,2}^{\text{sat}} + T_{n,3}^{\text{sat}}) + \mu_n^E E_{n,2}^{\text{sat}}(s_{\mathcal{A}}, \Xi[t]) \leq \mu_n^T T_n^{\text{loc}} + \mu_n^E E_n^{\text{loc}} \quad (6)$$

式(6)可变形为

$$R_n(s_{\mathcal{A}}, \Xi[t]) \geq \frac{(\mu_n^T + \mu_n^E p_n) W_n}{\mu_n^T (T_n^{\text{loc}} - T_{n,2}^{\text{sat}} - T_{n,3}^{\text{sat}}) + \mu_n^E (E_n^{\text{loc}} - E_{n,2}^{\text{sat}})} \quad (7)$$

进一步地, 根据式(1)可以得到

$$\sum_{j \in \mathcal{A}\setminus\{n\}; s_j = s_n} p_j g_{j,\text{sat}} \leq \frac{p_n g_{n,\text{sat}}[t]}{\frac{(\mu_n^E p_n + \mu_n^T) W_n}{2^{B(\mu_n^T(T_n^{\text{loc}} - T_{n,2}^{\text{sat}} - T_{n,3}^{\text{sat}}) + \mu_n^E(E_n^{\text{loc}} - E_{n,2}^{\text{sat}}))} - 1} - \sigma_0} \quad (8)$$

根据上述结论, 用户 n 的代价函数 u_n^0 的等价形式可以表示为

$$u_n(s_n, s_{\mathcal{A}\setminus\{n\}}, \Xi[t]) = \begin{cases} A_n[t], & s_n = 0 \\ J_n(s_{\mathcal{A}}, \Xi[t]), & s_n > 0 \end{cases} \quad (9)$$

可以看出, 式(5)和式(9)中用户的策略偏好是相同的。通过式(9), 用户只需感知每个信道中存在的干扰就可以独立选择卸载策略, 而不需要获取全局信息(即其他用户的策略)。因此, 本文通过将最小化卸载成本(式(5))转化为最小化信道干扰(式(9))来执行分布式解决方案。

接下来, 将事件 Ξ 扩展到随机和动态的环境(迭代所有事件)。对于每个用户 $n \in \mathcal{N}$, 在随机动态环境中, 其期望代价函数为

$$\bar{u}_n(s_n, s_{-n}) = \begin{cases} \bar{A}_n, & s_n = 0 \\ E_{\Theta}[J_n(s_n, s_{-n}, \Theta)], & s_n > 0 \end{cases} \quad (10)$$

其中, E_{Θ} 是期望算子, $\bar{A}_n = \frac{p_n \bar{g}_{n,\text{sat}}}{2^{\psi_n} - 1} - \sigma_0$, $\bar{g}_{n,\text{sat}}$ 是期望信道增益。根据式(10), 将随机博弈定义为 $\mathcal{G} = [\mathcal{N}, \Theta, \{S_n\}_{n \in \mathcal{N}}, \{\bar{u}_n\}_{n \in \mathcal{N}}]$, 其中, S_n 表示活跃用户 n 的策略集。每个活跃用户 n 独立选择其策略 s_n 以最小化其所经历的干扰, 因此博弈模型可以表示为

$$(\mathcal{G}): \min_{s_n \in S_n} \bar{u}_n(s_n, s_{-n}), \forall n \in \mathcal{N} \quad (11)$$

2.2 NE分析

定义1 \mathcal{G} 的NE。对于随机博弈 \mathcal{G} , 当且仅当没有用户可以通过偏离当前策略来降低其期望代价函数 \bar{u}_n , 即

$$\bar{u}_n(s_n^*, s_{\mathcal{A}\setminus\{n\}}^*) \leq \bar{u}_n(s_n, s_{\mathcal{A}\setminus\{n\}}^*), \forall n \in \mathcal{N}, \forall s_n \in S_n \quad (12)$$

则博弈结果是稳定的, 策略组合 $s_{\mathcal{A}}^* = [s_n^*]_{\forall n \in \mathcal{A}}$ 是纯策略纳什均衡。

定理 1 \mathcal{G} 是一个具有至少一个纯策略纳什均衡解的加权势博弈，全局最小化信道干扰的最优卸载策略是 \mathcal{G} 的最佳纯策略纳什均衡点。

证明 对于任意用户 $k \in \mathcal{A}$ ，其偏离所带来的代价函数的变化与势函数的变化成正比。具体地，构造一个势函数为

$$\begin{aligned} \bar{\Phi}(s_{\mathcal{A}}) &= \frac{1}{2} \sum_{n \in \mathcal{A}} \sum_{j \in \mathcal{A} \setminus \{n\}} \varphi_n \varphi_j p_n \bar{g}_{n,\text{sat}} p_j \bar{g}_{j,\text{sat}} \cdot \\ &\mathcal{L}_{\{s_j = s_n\}} \mathcal{L}_{\{s_n > 0\}} + \sum_{n \in \mathcal{A}} \varphi_n p_n \bar{g}_{n,\text{sat}} \bar{A}_n \mathcal{L}_{\{s_n = 0\}} \end{aligned} \quad (13)$$

其中， $\mathcal{L}_{\{\text{condition}\}}$ 为指示函数，当条件为真时， $\mathcal{L}_{\{\text{condition}\}} = 1$ ；反之， $\mathcal{L}_{\{\text{condition}\}} = 0$ 。式(13)的等价形式可以表示为

$$\begin{aligned} \bar{\Phi}(s_{\mathcal{A}}) &= \frac{1}{2} \left(\sum_{j \in \mathcal{A} \setminus \{k\}} p_k \bar{g}_{k,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_k\}} \mathcal{L}_{\{s_k > 0\}} + \right. \\ &\sum_{n \in \mathcal{A} \setminus \{k\}} p_n \bar{g}_{n,\text{sat}} p_k \bar{g}_{k,\text{sat}} \mathcal{L}_{\{s_k = s_n\}} \mathcal{L}_{\{s_n > 0\}} + \\ &\left. \sum_{n \in \mathcal{A} \setminus \{k\}} \sum_{j \in \mathcal{A} \setminus \{n,k\}} p_n \bar{g}_{n,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_n\}} \mathcal{L}_{\{s_n > 0\}} \right) + \\ &p_k \bar{g}_{k,\text{sat}} \bar{A}_k \mathcal{L}_{\{s_k = 0\}} + \sum_{n \in \mathcal{A} \setminus \{k\}} p_n \bar{g}_{n,\text{sat}} \bar{A}_n \mathcal{L}_{\{s_n = 0\}} \end{aligned} \quad (14)$$

此外，由于干扰的对称性可得

$$\begin{aligned} \sum_{j \in \mathcal{A} \setminus \{k\}} p_k \bar{g}_{k,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_k\}} \mathcal{L}_{\{s_k > 0\}} = \\ \sum_{n \in \mathcal{A} \setminus \{k\}} p_n \bar{g}_{n,\text{sat}} p_k \bar{g}_{k,\text{sat}} \mathcal{L}_{\{s_k = s_n\}} \mathcal{L}_{\{s_n > 0\}} \end{aligned} \quad (15)$$

根据式(14)和式(15)可得

$$\begin{aligned} \bar{\Phi}(s_{\mathcal{A}}) &= \sum_{j \in \mathcal{A} \setminus \{k\}} p_k \bar{g}_{k,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_k\}} \mathcal{L}_{\{s_k > 0\}} + \\ &p_k \bar{g}_{k,\text{sat}} \bar{A}_k \mathcal{L}_{\{s_k = 0\}} + \bar{\Psi}(s_{\mathcal{A} \setminus \{k\}}) \end{aligned} \quad (16)$$

其中，

$$\begin{aligned} \bar{\Psi}(s_{\mathcal{A} \setminus \{k\}}) &= \\ &\sum_{n \in \mathcal{A} \setminus \{k\}} p_n \bar{g}_{n,\text{sat}} \bar{A}_n \mathcal{L}_{\{s_n = 0\}} + \\ &\frac{1}{2} \sum_{n \in \mathcal{A} \setminus \{k\}} \sum_{j \in \mathcal{A} \setminus \{n,k\}} p_n \bar{g}_{n,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_n\}} \mathcal{L}_{\{s_n > 0\}} \end{aligned}$$

是与用户 k 的计算策略 s_k 无关的项。

进一步地，根据式(10)可以得到

$$\begin{aligned} \bar{u}_k(s_{k \rightarrow s_{-k}}) &= J_k(s_{\mathcal{A}}) \mathcal{L}_{\{s_k > 0\}} + \bar{A}_k \mathcal{L}_{\{s_k = 0\}} = \\ &\sum_{j \in \mathcal{A} \setminus \{k\}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_k\}} \mathcal{L}_{\{s_k > 0\}} + \bar{A}_k \mathcal{L}_{\{s_k = 0\}} \end{aligned} \quad (17)$$

因此，对于任意用户 k 及其 2 种不同的计算策

略 s_k 和 s_k^* ，有

$$\begin{aligned} \bar{\Phi}(s_{k \rightarrow s_{-k}}^*) - \bar{\Phi}(s_{k \rightarrow s_{-k}}) &= \\ &\sum_{j \in \mathcal{A} \setminus \{k\}} p_k \bar{g}_{k,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_k^*\}} \mathcal{L}_{\{s_k^* > 0\}} - \\ &\sum_{j \in \mathcal{A} \setminus \{k\}} p_k \bar{g}_{k,\text{sat}} p_j \bar{g}_{j,\text{sat}} \mathcal{L}_{\{s_j = s_k\}} \mathcal{L}_{\{s_k > 0\}} + \\ &p_k \bar{g}_{k,\text{sat}} \bar{A}_k \mathcal{L}_{\{s_k^* = 0\}} + p_k \bar{g}_{k,\text{sat}} \bar{A}_k \mathcal{L}_{\{s_k = 0\}} = \\ &p_k \bar{g}_{k,\text{sat}} (\bar{u}_k(s_{k \rightarrow s_{-k}}^*) - \bar{u}_k(s_{k \rightarrow s_{-k}})) \end{aligned} \quad (18)$$

式(18)表明，任意用户 $k \in \mathcal{A}$ 的策略偏离所引起的代价函数的变化与式(13)中定义的势函数的变化成正比。因此，根据势博弈理论^[16]， \mathcal{G} 是一个加权势博弈（权重因子为 $p_k \bar{g}_{k,\text{sat}}$ ），并且至少有一个纳什均衡解。势函数的任意全局或局部最小值构成了博弈 \mathcal{G} 的纯策略纳什均衡点^[16]。根据势函数与信道干扰之间的关系，可以得出使势函数最小的最优纳什均衡也可以使信道干扰最小。证毕。

接下来求解纳什均衡。在动态环境中，用户以分布式的方式到达纳什均衡点是具有挑战性的。可用的现有博弈论算法包括最佳响应算法^[16,24]、空间自适应算法等。但这些算法需要大量的信息交互进行用户策略更新，并且算法收敛时要求环境是静态的。为了得到所提博弈 \mathcal{G} 的纳什均衡，本文将随机学习^[25] 纳入动态环境下分布式算法的构建中。本文设计的智能随机学习算法如算法 1 所示。

算法 1 智能随机学习算法

初始化 设置迭代次数 $t = 0$ ，用户 n 的学习步长 b ，设置每个用户 $n \in \mathcal{N}$ 的初始混合策略概率向量为均匀分布 $\mathbf{v}_i^0 = \left(\frac{1}{M+1}, \dots, \frac{1}{M+1} \right)$ 。

- 1) 循环；
- 2) 迭代次数更新 $t = t + 1$ ；
- 3) 如果 $n \in \mathcal{A}'$ ，用户 n 根据其当前策略选择概率向量 \mathbf{v}_n^t 随机选择卸载策略 s_n^t ；否则，用户 $\mathcal{N} \setminus \mathcal{A}'$ 保持沉默，不采取任何行动；
- 4) 如果 $n \in \mathcal{A}'$ ，用户 n 根据式(9)评估其各自的代价值 \bar{u}_n^t ；否则，用户 $\mathcal{N} \setminus \mathcal{A}'$ 保持沉默，不采取任何行动；
- 5) 如果 $s_n^t > 0$ ，用户 n 根据式(9)评估其受到的干扰 J_n^t ；否则，用户 n 直接计算 A_n^t ；
- 6) 如果 $n \in \mathcal{A}'$ ，用户根据以下规则更新其下一个时隙的混合策略概率向量

$$\mathbf{v}_n^{t+1} = \mathbf{v}_n^t + bq_n^t(\mathbf{e}_{s_n^t} - \mathbf{v}_n^t) \quad (19)$$

其中, $0 < b < 1$ 是学习步长, $\mathbf{e}_{s_n^t}$ 是 $(M + 1)$ 维单位向量, q_n^t 是收到的动作奖励, 具体定义为 $q_n^t = 1 - \varepsilon_n \mu_n^t$; 否则, 用户 $\mathcal{N} \setminus \mathcal{M}$ 保持其策略选择概率向量不变, 即 $\mathbf{v}_n^{t+1} = \mathbf{v}_n^t$;

7) 直到所有用户不再调整各自的策略。

8) 输出混合策略概率向量 $\mathbf{v}_i^t, \forall n \in \mathcal{N}, t \geq 1$ 。

具体来说, 每个活跃用户充当学习自动机, 根据其策略空间的概率向量独立随机地选择计算策略, 然后根据动态环境中获得的反馈值更新其概率向量。每个用户最终都会通过不断与环境交互来调整到最适合自己的策略。所提算法是完全分布式的, 计算复杂度为 $\mathcal{O}(CN)^{[26]}$, 其中, C 是迭代次数。因此, 所提算法能够以较低的复杂度实现, 具有较好的适用性。

3 仿真分析

本节进行了仿真实验来验证所提算法在动态环境中的性能。考虑一个网络, 其中一组地面用户随机分布在卫星覆盖区域内的 $1\,000\text{ m} \times 1\,000\text{ m}$ 的方形区域中。对于每个用户 n , 其活跃概率 φ_n 在 $(0, 1]$ 内均匀分布, 时变信道服从莱斯分布, 随机衰落因子 β 服从单位均值的指数分布。其他仿真参数如表 2 所示^[11, 17]。

表 2 其他仿真参数

参数	值
卫星覆盖区域	1 000 m × 1 000 m
用户活跃概率 φ_n	(0, 1]
带宽 B/MHz	20
最大传输功率 p_f/mW	200
卫星高度 h/km	784
噪声功率 σ_0/dBm	-100
路径损耗指数 α	4
数据量大小 W_i/KB	5 000
本地计算 CPU 转数 $D_i^{\text{loc}}/\text{Megacycle}$	1 000
在轨计算 CPU 转数 $D_i^{\text{sat}}/\text{Megacycle}$	3 000
本地计算能力 $f_i^{\text{loc}}/\text{GHz}$	1
卫星计算能力 $f_i^{\text{sat}}/\text{GHz}$	12
计算能耗权重 μ_i^E	0 ~ 1
计算时延权重 μ_i^T	$1 - \mu_i^E$

以下仿真结果是通过 500 次单独实验进行平均而得到的。设置默认用户数量为 30, 信道数量

为 5。信道选择的收敛过程如图 2 所示。从图 2 可以看出, 在迭代开始时, 用户根据策略空间中的均匀分布随机选择信道。随着迭代的进行, 用户的信道选择概率不断变化, 并在 200 次迭代后收敛。信道 4 是用户计算卸载的最终选择, 选择其他任何策略的概率为 0。这验证了所提算法的收敛性。

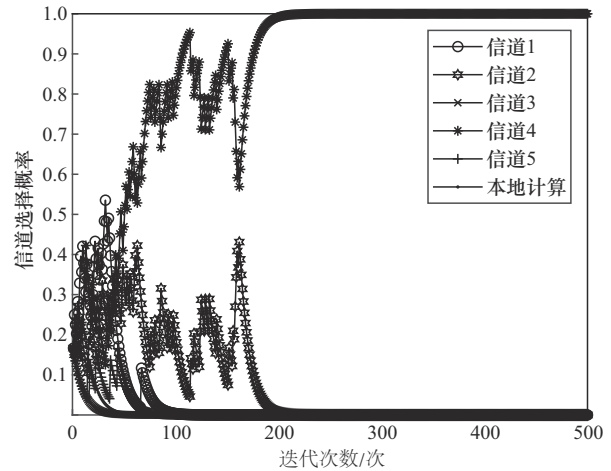


图 2 信道选择的收敛过程

学习步长 b 对性能的影响如图 3 所示, 设置 b 分别为 0.05、0.1、0.2、0.3、0.4。从图 3 可以看出, 随着 b 的增大, 算法收敛速度加快, 但只能得到非纳什均衡解, 性能较差。这是因为 b 是开发和探测过程的控制参数, b 的增加减少了算法探索策略选择的机会, 从而导致了纳什均衡解的低效性。但较小的 b 值会导致算法收敛过慢, 例如 $b = 0.05$ 时需要超过 500 次迭代才能收敛。因此, b 的取值需要在纳什均衡的收敛速度和精度之间进行折中选择。受图 3 启发, 本文选择 $b = 0.1$ 作为仿真参数。

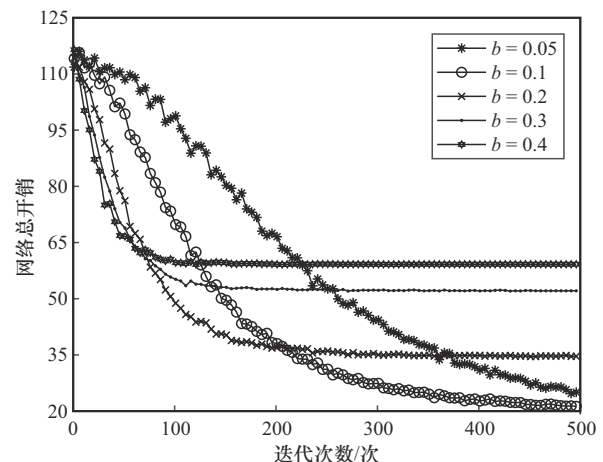


图 3 学习步长 b 对性能的影响

接下来,进一步验证所提算法的开销性能,并与3种基准算法,即随机算法、最优响应算法和交替方向乘子算法^[15]进行了比较。

网络总开销与用户数量之间的关系如图4所示。从图4可以看出,随着用户数量的增加,随机算法、最优响应算法和交替方向乘子算法的网络总开销呈明显增加趋势,反映出卸载用户之间的竞争加剧,从而导致同信道的干扰增加。而所提算法的网络总开销增加缓慢,表现出了较强的鲁棒性,即网络总开销不会随着用户数量的增加而激增。这是因为所提算法是完全分布式的,不需要大量的信息交互来进行策略更新,所以可以快速适应环境的动态变化,从而选择更好的接入策略,保持在较低的网络开销水平。与基准算法相比,所提算法显著降低了系统成本,并最接近全局最优。

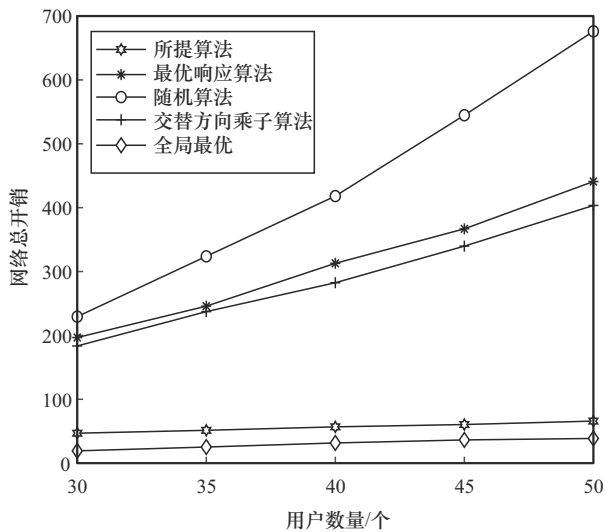


图4 网络总开销与用户数量之间的关系

网络总开销与信道数量之间的关系如图5所示。首先,可以直观地看到,每种算法的网络总开销随着信道数量的增加而减小,符合预期。这是因为更多的接入选择使同一信道上的用户分布数量减少,所以卸载用户之间的同信道干扰也随之降低。此外,所提算法的性能与图4所示一致,表现出了较强的鲁棒性。与随机算法、最优响应算法和交替方向乘子算法相比,所提算法均实现了最低的网络总开销,并且最接近全局最优,这证明了本文所提智能随机学习算法的有效性。

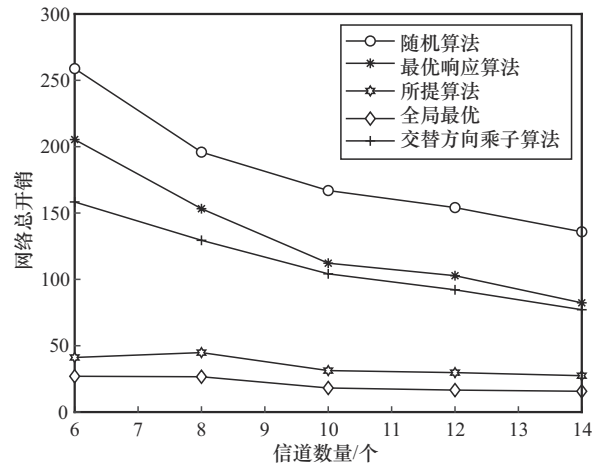


图5 网络总开销与信道数量之间的关系

4 结束语

为了解决动态环境下的LEO星座边缘计算卸载难题,本文考虑环境的高动态性和活跃用户的随机访问性,采用博弈论对星地多用户计算卸载问题进行了研究。首先,本文致力于实现系统开销的最小化,所提出的博弈问题被证明是一个加权势博弈,其中纳什均衡解将干扰降低到全局最低。然后,为了求解纳什均衡,设计了一种基于随机学习的完全分布式的智能随机学习算法。最后,通过数值仿真验证了相较于基准算法,所提算法具有较好的计算卸载性能和鲁棒性。本文工作专注于信道干扰和成本最小化,这可能无法保证系统的公平性。因此,未来可以致力于将频谱效率、公平性和用户需求共同考虑,以提高最终的用户体验质量。

参考文献:

- [1] CASSARÁ P, GOTTA A, MARCHESE M, et al. Orbital edge offloading on mega-LEO satellite constellations for equal access to computing[J]. IEEE Communications Magazine, 2022, 60(4): 32-36.
- [2] 赵亚飞, 闫冰, 孙耀华, 等. 低轨星座通导一体化: 现状、机遇和挑战[J]. 电信科学, 2023, 39(5): 90-100.
- [3] ZHAO Y F, YAN B, SUN Y H, et al. Communication and navigation integration for LEO constellations: status, opportunities, and challenges[J]. Telecommunications Science, 2023, 39(5): 90-100
- [4] WIGARD J, JUAN E, STANCZAK J, et al. Ubiquitous 6G service through non-terrestrial networks[J]. IEEE Wireless Communications, 2023, 30(6): 12-18.
- [5] AKYILDIZ I F, KAK A. The Internet of space things/CubeSats[J]. IEEE Network, 2019, 33(5): 212-218.
- [6] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.
- [7] ZHAO L, SUN W, SHI Y P, et al. Optimal placement of cloudlets for access delay minimization in SDN-based Internet of things networks[J]. IEEE Internet of Things Journal, 2018, 5(2): 1334-1344.

- [7] 佟明磊, 李松, 王晓湘. 移动边缘计算使能的星地网络 D2D 辅助任务卸载[J]. 北京邮电大学学报, 2024, 47(1): 25-30.
TONG M L, LI S, WANG X X. D2D-assisted task offloading in mobile edge computing-enabled satellite-terrestrial networks[J]. Journal of Beijing University of Posts and Telecommunications, 2024, 47(1): 25-30.
- [8] 唐琴琴, 谢人超, 刘旭, 等. 融合 MEC 的星地协同网络: 架构、关键技术与挑战[J]. 通信学报, 2020, 41(4): 162-181.
TANG Q Q, XIE R C, LIU X, et al. MEC enabled satellite-terrestrial network: architecture, key technique and challenge[J]. Journal on Communications, 2020, 41(4): 162-181.
- [9] 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.
- [10] CHENG N, LYU F, QUAN W, et al. Space/aerial-assisted computing offloading for IoT applications: a learning-based approach[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(5): 1117-1129.
- [11] BHATTACHERJEE D, KASSING S, LICCIARDELLO M, et al. In-orbit computing: an outlandish thought experiment? [C]//Proceedings of the 19th ACM Workshop on Hot Topics in Networks. New York: ACM Press, 2020: 197-204.
- [12] LI Q, WANG S G, MA X, et al. Service coverage for satellite edge computing[J]. IEEE Internet of Things Journal, 2022, 9(1): 695-705.
- [13] WANG Y J, ZHANG J X, ZHANG X, et al. A computation offloading strategy in satellite terrestrial networks with double edge computing[C]//Proceedings of the 2018 IEEE International Conference on Communication Systems (ICCS). Piscataway: IEEE Press, 2018: 450-455.
- [14] ZHU D L, LIU H T, LI T, et al. Deep reinforcement learning-based task offloading in satellite-terrestrial edge computing networks[C]//Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway: IEEE Press, 2021: 1-7.
- [15] 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.
- [16] MONDERER D, SHAPLEY L S. Potential games[J]. Games and Economic Behavior, 1996, 14(1): 124-143.
- [17] WANG Y X, YANG J, GUO X Y, et al. A game-theoretic approach to computation offloading in satellite edge computing[J]. IEEE Access, 2019, 8: 12510-12520.
- [18] WANG B, FENG T, HUANG D Y. A joint computation offloading and resource allocation strategy for LEO satellite edge computing system[C]//Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT). Piscataway: IEEE Press, 2020: 649-655.
- [19] WANG D Z, WANG W, KANG Y H, et al. Distributed data offloading in ultra-dense LEO satellite networks: a Stackelberg mean-field game approach[J]. IEEE Journal of Selected Topics in Signal Processing, 2023, 17(1): 112-127.
- [20] 菅欣柯, 徐飞, 宁临鹏. 低轨卫星网络基于 Stackelberg 博弈的任务卸载策略[J]. 西安工业大学学报, 2023, 43(4): 393-404.
JIAN X K, XU F, NING L P. Research on task offloading strategy based on Stackelberg game in LEO satellite network[J]. Journal of Xi'an Technological University, 2023, 43(4): 393-404.
- [21] CHEN X. Decentralized computation offloading game for mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(4): 974-983.
- [22] 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.
- [23] ABDI A, LAU W C, ALOUINI MS, et al. A new simple model for

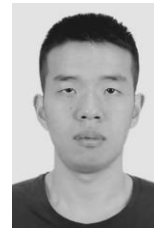
land mobile satellite channels: first- and second-order statistics[J]. IEEE Transactions on Wireless Communications, 2003, 2(3): 519-528.

- [24] CHEN X, JIAO L, LI W Z, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2016, 24(5): 2795-2808.
- [25] SASTRY P S, PHANSALKAR V V, THATHACHAR M A L. Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1994, 24(5): 769-777.
- [26] XU Y H, WANG J L, WU Q H, et al. Opportunistic spectrum access in cognitive radio networks: global optimization using local interaction games[J]. IEEE Journal of Selected Topics in Signal Processing, 2012, 6(2): 180-194.

[作者简介]



高玉芳 (1996-), 女, 山西长治人, 南京大学博士生, 主要研究方向为无线资源分配、边缘计算、卫星通信、博弈论等。



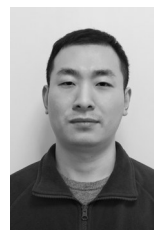
姬智 (1997-), 男, 河南洛阳人, 陆军工程大学博士生, 主要研究方向为无人机通信、资源分配、博弈论等。



赵康健 (1981-), 男, 上海人, 博士, 南京大学教授、博士生导师, 主要研究方向为空间网络体系架构与仿真、星地一体化网络等。



李文峰 (1975-), 男, 江苏常州人, 博士, 南京大学研究员, 主要研究方向为空间网络架构、卫星通信与网络、无线与移动通信等。



胡佩聪 (1991-), 男, 江苏镇江人, 63893 部队工程师, 主要研究方向为无线通信。