天地算力网络中的异构资源协同博弈

张雨童[1]，彭煜明[1]，邸博雅[1]，宋令阳[1,2]
（1. 北京大学区域光纤通信网与新型光通信系统国家重点实验室，北京 100871；2. 北京大学深圳研究生院信息工程学院，广东 深圳 518055）

摘 要：为解决多卫星天地算力网络中的星间资源博弈，围绕计算、频谱域资源管理问题，设计了一种天地异构资源协同博弈机制。每颗卫星搭载一项计算任务，各任务间彼此独立，依赖用户设备从环境中获取原始数据，通过竞争网络中的计算/频谱资源实现数据卸载与计算。为提供高速数据服务，提出基于多智能体强化学习的分布式算法，以协调星间异构资源竞争，实现系统时延最小化。仿真表明，与现有方案相比，所提算法可获得更低的系统时延。

关键词：天地算力网络；异构资源协同博弈；多智能体强化学习
中图分类号：TN927
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2023214

Heterogeneous resource cooperative game in space-ground computing power network

ZHANG Yutong[1], PENG Yuming[1], DI Boya[1], SONG Lingyang[1,2]
1. State Key Laboratory of Advanced Optical Communication Systems and Networks, Peking University, Beijing 100871, China
2. School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China

Abstract: To deal with the resource competition among satellites in the multi-satellite space-ground computing network, a space-ground heterogeneous resource cooperative game mechanism was designed in terms of the computing and spectrum domains. Each satellite published a computing task which was independent of other tasks and relied on UE to generate raw data. By competing the resources of user terminals and UE, the task offloading and processing was achieved. To provide real-time data services, a distributed scheme was proposed based on multi-agent reinforcement learning to coordinate the computing and spectrum resource competition among satellites, thereby minimizing the system latency. Simulation results indicated that, compared with the existing schemes, the proposed algorithm achieves a lower system latency by fully utilizing the computing and spectrum resources and coordinating the resource competition.

Keywords: space-ground computing power network, heterogeneous resource cooperative game, multi-agent reinforcement learning

0 引言

近年来，卫星通信作为下一代无线通信的关键技术之一[1]，其凭借覆盖范围广、通信距离远、不受地理条件限制等优点，广泛应用于各类场景中[2-3]。例如，在抢险救灾中，需要构建灾难现场的实况地图，从而给救援提供便利[4]。该任务需要大量用户设备（UE, user equipment）收集和上传现场数据。然而，灾难场景下地面通信受阻，可将卫星作为中继，将现场数据转发至地面站进行计算处理，形成天地一

收稿日期：2023−06−30；**修回日期**：2023−12−12
通信作者：宋令阳，lingyang.song@pku.edu.cn
基金项目：国家重点研发计划基金资助项目（No.2022YFE0111900）；湖南省科技创新计划基金资助项目（No.2022RC4024）；国家自然科学基金资助项目（No.62227809, No.61931019, No.62271012）；北京市自然科学基金资助项目（No.L212027, No.4222005）
Foundation Items: The National Key Research and Development Program of China (No.2022YFE0111900), The Science and Technology Innovation Program of Hunan Province (No.2022RC4024), The National Natural Science Foundation of China (No.62227809, No.61931019, No.62271012), The Beijing Natural Science Foundation (No.L212027, No.4222005)

体化网络[5]。

然而，由于频谱资源有限，天地一体化网络面临巨大的传输压力[6]，从而带来较高的系统时延，难以满足未来通信网络中的高速数据服务需求。针对这一问题，考虑将天地一体化网络与算力网络[7]相融合，形成天地算力网络。该网络将计算资源部署在网络边缘的 UE 及地面−卫星终端（TST, terrestrial-satellite terminal）上，并将部分数据卸载至 UE 及 TST 处进行计算，以实现数据压缩，从而缓解网络中的传输压力。为充分利用网络中各处的算力，进而满足海量数据传输需求，需要将网络中的计算域及频谱域资源进行联合协同管理，以实现系统时延最小化，提供高速数据服务。

现有工作分别对网络中的单一资源分配[8-11]和异构资源分配[12-13]进行了研究。文献[8]针对网络中的频谱资源设计了一种基于多智能体深度强化学习的资源分配方案。文献[11]研究了卫星集群网络中的传输资源分配，在保证数据速率的前提下最大程度地提高系统能量效率。文献[13]综合考虑空天地网络中的缓存、计算及传输资源，设计了一种联合资源分配算法，使网络能量效率最大化。文献[14]探究了卫星通信中的计算及传输资源分配问题，从而最小化系统时延。

然而，现有方案大多忽略了任务分布式部署这一实际问题。具体而言，由于不同卫星分属于不同系列[15]，如高分卫星[16]、动中通卫星[17]等，不同系列卫星之间彼此独立，且在服务类型、价格、资源部署等方面存在显著差异。因此，需要根据不同需求，将任务分布式部署在多颗不同的卫星上，如城市影像观测、地面环境监测等，这导致现有资源分配方案不再适用。为此，本文考虑了一种多卫星天地算力网络，每颗卫星分属不同系列，根据星间服务类型、资源部署等方面的特性差异，每颗卫星上各自搭载一项特定的计算任务（例如，得益于国家高分辨率对地观测系统，高分卫星常搭载观测城市影像等任务，而动中通卫星更常用于卫星通信），依靠 UE 从环境中获取原始数据。例如，对于城市影像观测任务而言，原始数据可能为照片、视频等图像数据。对于地面环境监测任务而言，原始数据可能为温湿度、PM2.5 等传感数据。UE 获取到的原始数据经由 TST 发送至卫星。数据传输过程中的各设备（UE、TST 及卫星）分别对部分原始数据进行计算，例如，对城市影像观测任务中的原始图像数据进行特征提取等；对地面环境监测任务中的温湿度等原始传感数据进行统计拟合等。最终将全部计算结果，即图像特征值或环境传感数据统计值，汇聚到地面站，生成城市影像或环境状况分布图，进行任务数据交付。

考虑到多卫星天地算力网络中的星间博弈和异构资源管理，该网络面临以下挑战。1) 由于各颗卫星分属不同系列，其上搭载的任务之间彼此独立，信息交互不完全，为最小化自身时延，各个任务将对 TST 和 UE 上的计算域及频谱域资源展开竞争，从而形成天地异构资源协同博弈（SHRCG, space-ground heterogeneous resource cooperative game）。这导致传统算法不再适用于此情况，亟须设计一种分布式算法来解决这一问题。2) 与传统地面移动通信网络不同，天地算力网络中包含卫星通信和地面通信，形成立体多维网络架构。任务处理时延同时受限于卫星及地面上的频谱及计算资源，这导致天地间的立体资源耦合，为资源分配带来挑战。3) 由于各个 TST、UE 上的计算及频谱资源有限，任务卸载策略受限于网络中所能调用的资源，因此任务卸载、计算和频谱资源分配相互耦合，给上述分布式算法设计带来额外的困难。

为解决上述挑战，本文主要贡献如下。

1) 建立了一种由多颗卫星、TST 及 UE 组成的天地算力网络，每颗卫星上搭载一项计算任务，不同任务间信息交互不完全，为最小化自身时延，需竞争 TST 及 UE 上的计算/频谱资源，形成天地异构资源协同博弈。

2) 针对上述博弈，提出了一种基于多智能体强化学习（MARL, multi-agent reinforcement learning）的 SHRCG 算法，以协调任务间的计算/频谱资源竞争，分布式地最小化系统时延，并对所提算法的特性进行了分析。

3) 仿真结果表明，本文所提算法能够协调任务间资源竞争，实现系统时延最小化，并揭示了不同设备（卫星、TST 或 UE）的数量对系统时延及算法收敛速度的影响，以及算法复杂度和系统时延精度之间的权衡关系。

1 系统模型及问题定义

本节首先对天地算力网络进行简要介绍，并依次构建其中的任务计算及传输模型，进而定义系统时延最小化问题。符号定义如表 1 所示。

表 1　　符号定义

符号	定义
L	卫星总数
N	TST 总数
M	每个 TST 下的 UE 总数
$l^{n,m}$	UE m-TST n-卫星 l 链路上的子任务
$\lambda_{\mathrm{UE}}^{n,m}(l)$	子任务 $l^{n,m}$ 的数据产生速率
$x_{\mathrm{TST}}^{n,m}(l)$	对于子任务 $l^{n,m}$，在 TST n 上计算的原始数据占其接收到的原始数据总数的百分比
$x_{\mathrm{UE}}^{n,m}(l)$	对于子任务 $l^{n,m}$，在 UE m 上计算的原始数据占其产生的原始数据总数的百分比
$\phi_{\mathrm{TST}}^{n,m}(l)$	子任务 $l^{n,m}$ 的卫星-TST 传输资源
$\phi_{\mathrm{UE}}^{n,m}(l)$	子任务 $l^{n,m}$ 的 TST-UE 传输资源
$\theta_{\mathrm{SAT}}^{n,m}(l)$	子任务 $l^{n,m}$ 的卫星计算资源
$\theta_{\mathrm{TST}}^{n,m}(l)$	子任务 $l^{n,m}$ 的 TST 计算资源
$\theta_{\mathrm{UE}}^{n,m}(l)$	子任务 $l^{n,m}$ 的 UE 计算资源
$t_{\mathrm{SAT}}^{n,m}(l)$	子任务 $l^{n,m}$ 在卫星上的数据计算时间
$t_{\mathrm{TST}}^{n,m}(l)$	子任务 $l^{n,m}$ 在 TST 上的数据计算时间
$t_{\mathrm{UE}}^{n,m}(l)$	子任务 $l^{n,m}$ 在 UE 上的数据计算时间
$\tau_{\mathrm{TST}}^{n,m}(l)$	子任务 $l^{n,m}$ 从 TST 到卫星的数据传输时间
$\tau_{\mathrm{UE}}^{n,m}(l)$	子任务 $l^{n,m}$ 从 UE 到 TST 的数据传输时间
$T^{n,m}(l)$	子任务 $l^{n,m}$ 的时延
$T(l)$	任务 l 的时延
T	系统时延
ρ	数据压缩率

1.1　场景描述

如图 1 所示，本文考虑一个天地算力网络，其中包括 L 颗卫星，N 个 TST，以及 MN 个 UE。其中，每颗卫星分属不同系列，分别通过回传链路将待计算数据转发至计算能力较强的地面站，进行处理交付。为简化模型，不失一般性[18]，本文假设卫星到地面站数据传输时间固定为 t_c，因此，将卫星和地面站看作一个整体，为了便于描述，下文中统一采用卫星作为指代。该假设易于拓展到星地传输时延不固定的场景，本文将在未来工作中考虑该场景下的天地异构资源协同博弈机制设计。不同设备之间采用无线通信，具体来说，每个 TST 可经由无线链路连接到任意多颗卫星，并为 M 个 UE 提供服务；每个 UE 通过无线传输与一个且仅一个 TST 相连。每个 UE 和 TST 均具有一定的计算能力。

每颗卫星 l 上搭载一项且仅一项任务，依靠 UE 产生原始数据。为简化描述，将卫星 l 上搭载的任务表示为任务 l。假设所有任务均为比特间独立，即每个任务 l 产生的数据可以被任意地划分为多个部分，分别在不同设备上进行处理。由于卫星分属不同系列，任务之间相互独立，信息交互不完全，各个任务分布式决定：1) 如何将任务 l 卸载到不同设备，即任务卸载策略；2)每个设备分配多少计算和频谱资源，用于处理任务 l，即资源分配策略。每个任务仅考虑将自身时延最小化。

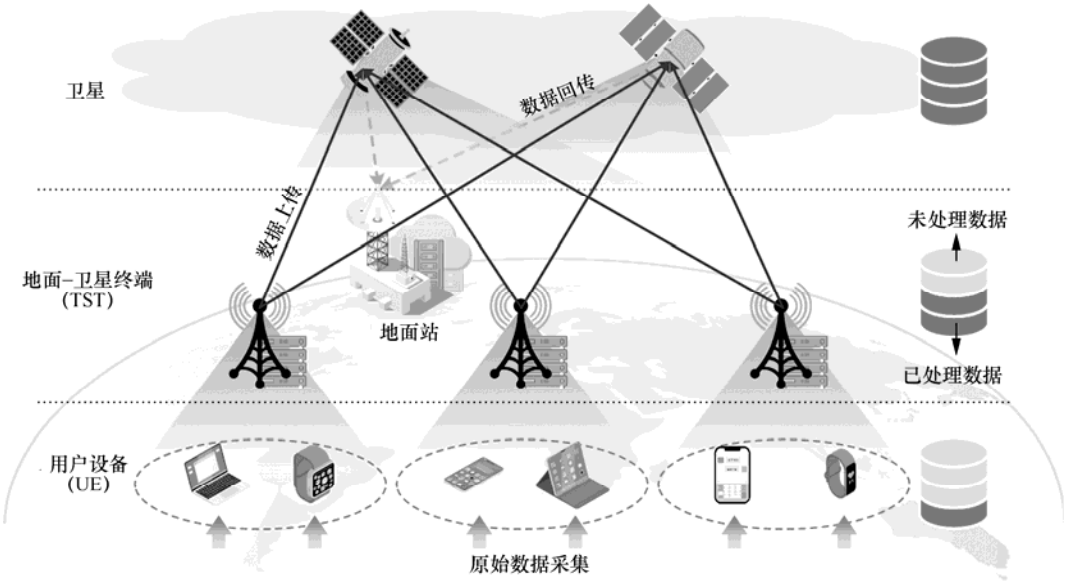


图 1　多卫星天地算力网络

为此，不同任务之间将竞争 TST 和 UE 上有限的计算和频谱资源，从而形成天地异构资源协同博弈架构。

当卫星下发任务卸载及资源分配策略后，TST 和 UE 将相应地进行数据计算及传输。具体地，对于每个任务 l，每个 UE 产生原始数据，并对部分数据进行计算，将计算结果及剩余原始数据传输给所连接的 TST。接收到来自 UE 的数据后，TST 依据卫星下发的任务卸载及资源分配策略进行部分数据计算，并将计算结果和剩余原始数据传输到卫星 l。卫星 l 将完成剩余原始数据的计算，并将全部计算结果汇总、交付。

1.2 任务计算和传输模型

每个任务 l 的原始数据由多个 UE 产生。为便于描述，本文将其中一个 UE 所产生数据的处理过程称为任务 l 的子任务，也就是将 UE m-TST n-卫星 l 链路上的数据处理称为子任务 $l^{n,m}$。在 UE m 处产生原始数据后，子任务 $l^{n,m}$ 的处理过程分为 5 个阶段：UE 处的数据计算、到 TST 的数据传输、TST 处的数据计算、到卫星的数据传输、卫星处的数据计算。经由逐层处理，当所有子任务的全部数据处理完毕，计算结果在卫星 l 处汇聚时，任务 l 完成。

1.2.1 数据计算模型及计算资源分配

本文采用 CPU 频率表征各个设备上的计算资源。令 θ_{SAT}、θ_{TST}、和 θ_{UE} 分别表示每颗卫星、TST、UE 上搭载的 CPU 频率，即每秒最大处理周期数。每比特原始数据计算所需周期数记作 Λ。

当 UE m 上用于处理子任务 $l^{n,m}$ 的计算资源为 $\theta_{\mathrm{UE}}^{n,m}(l)$ 时，各设备上的数据计算时间可以表示为

$$t_{\mathrm{UE}}^{n,m}(l)=\frac{\lambda_{\mathrm{UE}}^{n,m}(l)x_{\mathrm{UE}}^{n,m}(l)\Lambda}{\theta_{\mathrm{UE}}^{n,m}(l)} \tag{1}$$

其中，$x_{\mathrm{UE}}^{n,m}(l)$ 表示在 UE m 上进行本地计算的原始数据所占百分比，$\lambda_{\mathrm{UE}}^{n,m}(l)$ 表示子任务 $l^{n,m}$ 的数据产生速率。计算资源 $\theta_{\mathrm{UE}}^{n,m}(l)$ 需要满足

$$\sum_{l=1}^{L}\theta_{\mathrm{UE}}^{n,m}(l)\leqslant\theta_{\mathrm{UE}} \tag{2}$$

每时间单元内，TST n 从 UE m 处接收到的数据包括 UE m 处的计算结果 $\rho\lambda_{\mathrm{UE}}^{n,m}(l)x_{\mathrm{UE}}^{n,m}(l)$ 和剩余原始数据 $\lambda_{\mathrm{UE}}^{n,m}(l)(1-x_{\mathrm{UE}}^{n,m}(l))$，因此，原始数据占所有到达数据之比为 $\dfrac{1-x_{\mathrm{UE}}^{n,m}(l)}{1-x_{\mathrm{UE}}^{n,m}(l)+\rho x_{\mathrm{UE}}^{n,m}(l)}$。则 TST n 从 UE m 处的原始数据到达速率可以表示为

$$\lambda_{\mathrm{TST}}^{n,m}(l)=\phi_{\mathrm{UE}}^{n,m}(l)\frac{1-x_{\mathrm{UE}}^{n,m}(l)}{1-x_{\mathrm{UE}}^{n,m}(l)+\rho x_{\mathrm{UE}}^{n,m}(l)} \tag{3}$$

其中，$\phi_{\mathrm{UE}}^{n,m}(l)$ 表示 UE 和 TST 之间子任务 $l^{n,m}$ 的无线传输速度，ρ<1 表示计算前后数据大小的压缩率[19]。令 $x_{\mathrm{UT}}^{n,m}(l)$ 表示在 TST n 上进行处理的原始数据占到达该设备的所有原始数据之比，则在 TST n 处的计算时间为

$$t_{\mathrm{TST}}^{n,m}(l)=\frac{\lambda_{\mathrm{TST}}^{n,m}(l)x_{\mathrm{TST}}^{n,m}(l)\Lambda}{\theta_{\mathrm{TST}}^{n,m}(l)} \tag{4}$$

其中，$\theta_{\mathrm{TST}}^{n,m}(l)$ 表示 TST n 用于处理子任务 $l^{n,m}$ 的计算资源，需要满足

$$\sum_{l=1}^{L}\sum_{m=1}^{M}\theta_{\mathrm{TST}}^{n,m}(l)\leqslant\theta_{\mathrm{TST}} \tag{5}$$

卫星接收到来自 TST 的数据后，对剩余原始数据进行处理。卫星 l 的原始数据到达速度为

$$\lambda_{\mathrm{SAT}}^{n,m}(l)=\phi_{\mathrm{UT}}^{n,m}(l)\frac{1-x_{\mathrm{TST}}^{n,m}(l)}{1-x_{\mathrm{TST}}^{n,m}(l)+\rho x_{\mathrm{TST}}^{n,m}(l)+\dfrac{\rho x_{\mathrm{UE}}^{n,m}(l)}{1-x_{\mathrm{UE}}^{n,m}(l)}} \tag{6}$$

其中，$\phi_{\mathrm{UE}}^{n,m}(l)$ 表示子任务 $l^{n,m}$ 在 UE 和卫星之间的无线传输速度。卫星 l 用于处理各项子任务 $l^{n,m}$ 的计算资源 $\theta_{\mathrm{SAT}}^{n,m}(l)$ 不能超过卫星 l 所拥有的总计算资源，即

$$\sum_{n=1}^{N}\sum_{m=1}^{M}\theta_{\mathrm{SAT}}^{n,m}(l)\leqslant\theta_{\mathrm{SAT}} \tag{7}$$

在卫星 l 处的计算时间为

$$t_{\mathrm{SAT}}^{n,m}(l)=\frac{\lambda_{\mathrm{SAT}}^{n,m}(l)\Lambda}{\theta_{\mathrm{SAT}}^{n,m}(l)} \tag{8}$$

1.2.2 数据传输模型及频谱资源分配

将每颗卫星-TST、TST-UE 的频谱资源分别表示为 φ_{TST} 和 φ_{UE}，分别划分为 J_{TST} 和 J_{UE} 个正交子信道，每个子信道的数据传输容量，即每时间单元可传输的最大比特数为

$$\phi_{\mathrm{TST}}=\frac{\varphi_{\mathrm{TST}}}{J_{\mathrm{TST}}}\mathrm{lb}\left(1+\frac{G_{\mathrm{TST}}P_{\mathrm{TST}}D^{-\alpha}}{\sigma_{\mathrm{TST}}^{2}}\right) \tag{9}$$

$$\phi_{\mathrm{UE}}=\frac{\varphi_{\mathrm{UE}}}{J_{\mathrm{UE}}}\mathrm{lb}\left(1+\frac{G_{\mathrm{UE}}P_{\mathrm{UE}}d^{-\alpha}}{\sigma_{\mathrm{UE}}^{2}}\right) \tag{10}$$

其中，G_{TST} 和 G_{UE} 分别表示 TST 和 UE 的天线发射增益，P_{TST} 和 P_{UE} 分别表示 TST 和 UE 的发射功率。

D 表示卫星距地面高度，d 表示 TST 与 UE 之间的距离。卫星-TST 和 TST-UE 链路上的加性白噪声分别服从$\mathcal{CN}(0,\sigma_{\mathrm{UT}}^2)$和$\mathcal{CN}(0,\sigma_{\mathrm{UE}}^2)$分布。

从 UE m 传输到它所连接的 TST n 的数据包括剩余原始数据和 UE m 处的计算结果，子任务 $l^{n,m}$ 所占用的 UE m 和 TST n 间的无线传输资源，即每时间单元可传输的最大比特数 $\phi_{\mathrm{UE}}^{n,m}(l)=\phi_{\mathrm{UE}}J_{\mathrm{UE}}^{n,m}(l)$，其中 $J_{\mathrm{UE}}^{n,m}(l)$ 表示分配给子任务 $l^{n,m}$ 的子信道数量，需要满足

$$\sum_{l=1}^{L}\sum_{m=1}^{M}J_{\mathrm{UE}}^{n,m}(l)\leqslant J_{\mathrm{UE}} \tag{11}$$

从 UE m 到 TST n 的数据传输时间可以表示为

$$\tau_{\mathrm{UE}}^{n,m}(l)=\frac{\rho\lambda_{\mathrm{UE}}^{n,m}(l)x_{\mathrm{UE}}^{n,m}(l)+\lambda_{\mathrm{UE}}^{n,m}(l)(1-x_{\mathrm{UE}}^{n,m}(l))}{\phi_{\mathrm{UE}}^{n,m}(l)} \tag{12}$$

对于子任务 $l^{n,m}$，TST n 首先对部分接收到的原始数据进行计算，而后将剩余的原始数据、UE m 和 TST n 处的计算结果上传给卫星 l，所需传输的数据量为 $\rho\lambda_{\mathrm{TST}}^{n,m}(l)x_{\mathrm{TST}}^{n,m}(l)+\lambda_{\mathrm{TST}}^{n,m}(l)(1-x_{\mathrm{TST}}^{n,m}(l))\beta_{\mathrm{TST}}^{n,m}(l)$，其中 $\beta_{\mathrm{TST}}^{n,m}(l)$ 是 UE m 处的计算结果，可以表示为

$$\beta_{\mathrm{TST}}^{n,m}(l)=\phi_{\mathrm{UE}}^{n,m}(l)\frac{\rho x_{\mathrm{UE}}^{n,m}(l)}{1-x_{\mathrm{UE}}^{n,m}(l)+\rho x_{\mathrm{UE}}^{n,m}(l)} \tag{13}$$

因此，TST n 到卫星 l 的传输时间可以表示为

$$\tau_{\mathrm{TST}}^{n,m}(l)=\frac{\rho\lambda_{\mathrm{TST}}^{n,m}(l)x_{\mathrm{TST}}^{n,m}(l)+\lambda_{\mathrm{TST}}^{n,m}(l)(1-x_{\mathrm{TST}}^{n,m}(l))\beta_{\mathrm{TST}}^{n,m}(l)}{\phi_{\mathrm{TST}}^{n,m}(l)}+t_c \tag{14}$$

其中，$\phi_{\mathrm{TST}}^{n,m}(l)=\phi_{\mathrm{TST}}J_{\mathrm{TST}}^{n,m}(l)$ 表示子任务 $l^{n,m}$ 所占用的 TST n 和卫星 l 间的无线传输资源，$J_{\mathrm{TST}}^{n,m}(l)$ 表示分配给子任务 $l^{n,m}$ 的子信道数量。分配到各个子任务的子信道数量应小于子信道总数量，即

$$\sum_{n=1}^{N}\sum_{m=1}^{M}J_{\mathrm{TST}}^{n,m}(l)\leqslant J_{\mathrm{TST}} \tag{15}$$

1.3　任务数据流模型

如图 2 所示，本节首先建立子任务 $l^{n,m}$ 的数据流模型，并据此分别定义子任务时延和任务时延。

为便于描述，本文将任务处理过程划分为多个等长的时间单元。在每一设备上，数据计算和传输并行处理。因此，对于每个时间单元生成的原始数据，子任务 $l^{n,m}$ 的总处理时间 $T_{\mathrm{sum}}^{n,m}(l)$ 可表示为

$$\begin{aligned}T_{\mathrm{sum}}^{n,m}(l)=&\max\{t_{\mathrm{UE}}^{n,m}(l),\tau_{\mathrm{UE}}^{n,m}(l)\}+\\&\max\{t_{\mathrm{TST}}^{n,m}(l),\tau_{\mathrm{TST}}^{n,m}(l)\}+t_{\mathrm{SAT}}^{n,m}(l)\end{aligned} \tag{16}$$

在高负荷状态下，子任务处理可以描述为一个流水线。当待计算或待传输的数据没有到达当前设备时，该设备处于空闲状态。用于等待数据到达的时间称为空闲时间。其中，最长耗时阶段（包括原始数据产生）中不存在空闲时间，流水线的时延主要取决于该阶段的数据计算/传输时间。针对一个时间单元产生的数据，最长阶段耗时可以表示为

$$\begin{aligned}T_{\max}^{n,m}(l)=\max\Bigg\{&\frac{1}{\lambda_{\mathrm{UE}}^{n,m}(l)},t_{\mathrm{UE}}^{n,m}(l),\tau_{\mathrm{UE}}^{n,m}(l),\\&t_{\mathrm{TST}}^{n,m}(l),\tau_{\mathrm{TST}}^{n,m}(l),t_{\mathrm{SAT}}^{n,m}(l)\Bigg\}\end{aligned} \tag{17}$$

如图 2 所示，对于 P 个时间单元内产生的原始数据，子任务 $l^{n,m}$ 的时延为 $(P-1)T_{\max}^{n,m}(l)+T_{\mathrm{sum}}^{n,m}(l)$。当 P 趋近于无穷时，处理每时间单元内产生的原始数据所需时间，即子任务时延，可以表示为

$$T^{n,m}(l)=\lim_{P\to\infty}\frac{(P-1)T_{\max}^{n,m}(l)+T_{\mathrm{sum}}^{n,m}(l)}{P}=T_{\max}^{n,m}(l) \tag{18}$$

式(18)表明，子任务时延约等于最长耗时阶段

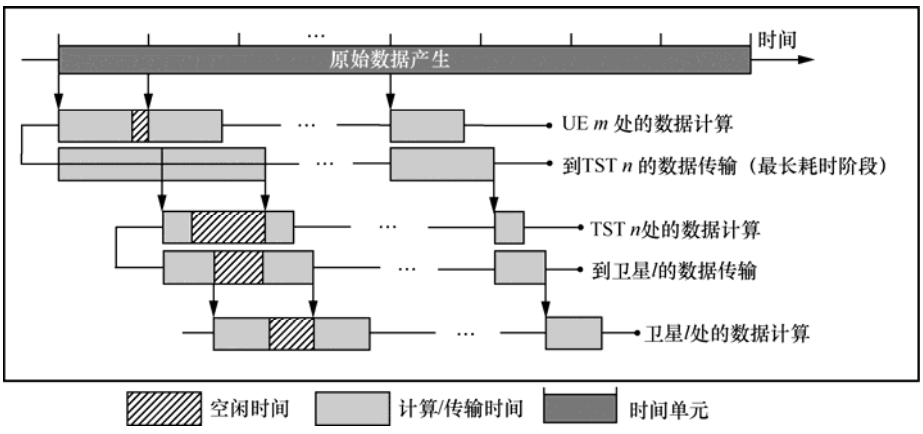


图 2　子任务 $l^{n,m}$ 数据流模型

的数据计算/传输时间。由于任务 l 的所有子任务均为并行处理，任务 l 的时延取决于所有子任务的最长时延，即

$$T(l)=\max_{n,m} T^{n,m}(l) \tag{19}$$

由于 L 颗卫星同时发布任务，天地算力系统延迟取决于最大任务时延，即

$$T=\max_{l} T(l) \tag{20}$$

2 天地异构资源协同博弈算法设计

考虑到天地算力网络的时变特性，如不稳定的无线环境，网络中的任务卸载和资源分配策略需要采用一种在线的方法进行优化，因此，本文设计了一种基于 MARL 的机制。具体地，本节首先定义了一个系统时延最小化问题，并将其重定义为一个 MARL 问题，并设计一种基于 MARL 的 SHRCG 算法，以使 L 颗卫星上的任务实现任务卸载和资源分配策略的分布式优化，从而最小化系统时延。

2.1 问题定义

考虑到各颗卫星之间相互独立，每颗卫星分布式地调整任务卸载和资源分配策略，从而最小化系统时延，即

$$\begin{aligned}
&\min_{\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}} \ T\\
&\text{s.t. } \{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}_1=\cdots=\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}_L=\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}\\
&\quad \sum_{l=1}^{L}\theta_{\mathrm{UE}}^{n,m}(l)\leqslant\theta_{\mathrm{UE}}\\
&\quad \sum_{l=1}^{L}\sum_{m=1}^{M}\theta_{\mathrm{TST}}^{n,m}(l)\leqslant\theta_{\mathrm{TST}}\\
&\quad \sum_{n=1}^{N}\sum_{m=1}^{M}\theta_{\mathrm{SAT}}^{n,m}(l)\leqslant\theta_{\mathrm{SAT}}\\
&\quad \sum_{l=1}^{L}\sum_{m=1}^{M}J_{\mathrm{UE}}^{n,m}(l)\leqslant J_{\mathrm{UE}}\\
&\quad \sum_{n=1}^{N}\sum_{m=1}^{M}J_{\mathrm{TST}}^{n,m}(l)\leqslant J_{\mathrm{TST}}
\end{aligned} \tag{21}$$

其中，$\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}_l$ 表示各个任务的任务卸载与资源分配策略。各个变量间的耦合关系分析如下。

1) 不同任务间的耦合。由于各颗卫星之间相互独立，每颗卫星上搭载一项任务，并分布式地调整任务卸载和资源分配策略。在该分布式决策过程中，为协调任务间的资源竞争，每颗卫星会将其他任务的历史决策作为指导，预测当前其他任务的决策。因此，需要保证搭载不同卫星所做出的资源分配策略与其他卫星保持一致，即 $\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}_1=\cdots=\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}_L=\{\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}\}$，以避免各个任务的资源分配策略产生冲突，从而最小化系统时延。

2) 不同设备间的耦合。每个任务都可以分割成不同的部分，分别在不同的设备（卫星、TST 或 UE）上进行处理，但整个网络的资源有限，任务卸载受限于相对应的设备所能调用的资源。所有任务所需的计算或频谱资源不应超过每个设备拥有的总资源，因此，不同设备之间的任务卸载与资源分配是相互耦合的。

2.2 基于学习的问题重定义

典型的 MARL 包含系统环境和多个智能体，通过环境和智能体之间的持续交互来选择最优的动作。在每次迭代中，每个智能体观察环境的当前状态，选择一个合适的动作并执行，系统状态随之发生转移，状态转移的好坏由一个强化信号（即奖励）表示。每个智能体倾向于选择可以使强化信号的长期累计值最大化的动作[20-21]。

天地算力网络中，每个任务的决策者被视为一个智能体，除此以外的一切被视为环境，即与其他任务的任务卸载和资源分配相关信息。任务发布后，每个智能体进行多次迭代，做出一系列任务卸载和资源分配决策，以进行试错。在每次迭代中，智能体观察环境状态，选择一个动作，并获得相应的奖励。当系统时延低于最大可容忍的系统时延且达到最大迭代次数 $i_{\max}$ 时，则将当前所选动作视为最优方案，用于最小化系统时延。对于同颗卫星，两次任务发布之间的时间间隔一般远大于获得最优方案所需的时间[22]。本节将依次定义天地算力网络的状态空间、动作空间和奖励函数。

1) 状态空间。状态空间包括二进制变量 β 和所有可能的数据产生速率，表示为 $\boldsymbol{\mathcal{S}}=[\beta,\boldsymbol{\lambda}_{\mathrm{UE}}]$。其中，二进制变量 β 用于指示当前系统时延是否满足预先确定的系统时延要求。具体来说，若执行新选择的动作所得到的系统时延低于最大可容忍的系统时延，则 $\beta=1$；反之，$\beta=0$。$\boldsymbol{\lambda}_{\mathrm{UE}}=\{\lambda_{\mathrm{UE}}^{n,m}(l)\,|\forall\, l,m,n\}$ 表示系统中各个 UE 的数据产生速率的集合。

2) 动作空间。为了联合描述天地算力网络中的任务卸载和资源分配，动作均由两部分组成，即任务卸载策略和资源分配策略。具体地，各个智能体可以获得的动作集合表示为 $\boldsymbol{\mathcal{A}}_1,\cdots,\boldsymbol{\mathcal{A}}_L$，其中，每个集合含有智能体 l 可能执行的动作，即 $\boldsymbol{\mathcal{A}}_l=\{\boldsymbol{a}_l\}=$

$\{[\boldsymbol{x}(l),\ \boldsymbol{\theta}(l),\ \boldsymbol{\phi}(l)]\}$。其中 $\boldsymbol{x}(l)=\{x_{\mathrm{UE}}^{n,m}(l),\ x_{\mathrm{TST}}^{n,m}(l)\,|\forall m,n\}$表示智能体 l 执行动作 $\boldsymbol{a}_l$ 时采用的任务卸载策略，$\boldsymbol{\theta}(l)=\{\theta_{\mathrm{UE}}^{n,m}(l),\ \theta_{\mathrm{TST}}^{n,m}(l),\ \theta_{\mathrm{SAT}}^{n,m}(l)\,|\forall m,n\}$和$\boldsymbol{\phi}(l)=\{\phi_{\mathrm{UE}}^{n,m}(l),\ \phi_{\mathrm{TST}}^{n,m}(l)\,|\forall m,n\}$表示智能体 l 执行动作 $\boldsymbol{a}_l$ 时采用的计算/传输资源分配策略。

3) 奖励函数。每一次迭代中，在当前状态 S 下，每个智能体 l 执行完动作 $\boldsymbol{a}_l$ 后，将会得到一个奖励 $R(S,\ \boldsymbol{a}_l)$。通常情况下，该奖励函数与目标函数相关。考虑到优化问题的目标是最小化系统时延，而 MARL 的目标是将奖励最大化，因此奖励应与系统时延呈负相关。为此，将即时奖励定义为归一化的系统时延，即

$$R=\frac{T_{\mathrm{local}}-T\left(S,\boldsymbol{a}_l\right)}{T_{\mathrm{local}}} \tag{22}$$

其中，T_{local} 是所有任务在 UE 本地执行时的系统时延，$T(S,\ \boldsymbol{a}_l)$是当前状态 S 下的实际系统时延，如式(20)所示。

通过上述定义，系统时延最小化问题式(21)可以重定义为一个马尔可夫决策过程[22]（MDP, Markov decision process）。例如，当数据传输速率 $\boldsymbol{\lambda}_{\mathrm{UE}}$ 被离散化为 K 个等级，其马尔可夫链如图 3 所示，其中 $p_{1,0}^{k,0}$ 表示从状态 $S_{1,0}=\{\beta=0,\boldsymbol{\lambda}_{\mathrm{UE},1}\}$到状态 $S_{k,0}=\{\beta=0,\boldsymbol{\lambda}_{\mathrm{UE},k}\}$的状态转移概率。具体地，在每次迭代中，智能体观察环境状态，决定问题式(21)中的优化变量，即任务卸载策略和资源分配方案。由于二进制变量 β 可能随系统时延的变化而变化，执行操作后，环境将从一种状态转换到另一种状态。随后，智能体将收到相应的奖励，用于表示动作选择的质量。

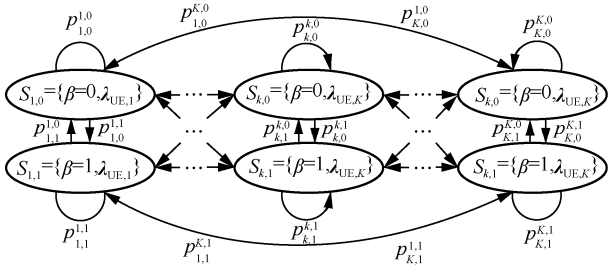


图 3　天地异构资源协同博弈中的马尔可夫链

2.3　基于 Q-learning 的天地异构资源协同博弈

联合动作学习者 Q-learning[23-24]是一种经典的 MARL 算法，它将强化学习（RL, reinforcement learning）与均衡学习法相结合，使多个智能体共同学习状态到动作的映射。具体来说，每个智能体学习其他智能体行为选择的平稳分布，建立显示模型，以指导自身的行为选择。

2.3.1　算法初始化

为将状态空间和动作空间构造为离散集，本文将数据产生速率、资源分配方案和任务分配策略离散化。进一步地，为了减小动作空间$\boldsymbol{\mathcal{A}}_l$的大小，一些不实际的操作将被从动作空间中删除，从而避免冗余。具体地，对于一个子任务 $l^{n,m}$，如果所有数据都在本地处理，则 TST 处的任务分配比必须为零。如果某个设备没有参与该子任务的数据处理，则该设备不应为子任务 $l^{n,m}$ 分配任何计算资源。即如果动作向量 $\boldsymbol{a}_l$ 包含以下状态之一，则需从动作空间中删除该动作向量：1) $x_{\mathrm{UE}}^{n,m}(l)=1$ 且 $x_{\mathrm{TST}}^{n,m}(l)\neq 0$；2) $x_{\mathrm{UE}}^{n,m}(l)=0$ 且 $\theta_{\mathrm{UE}}^{n,m}(l)\neq 0$；3) $x_{\mathrm{TST}}^{n,m}(l)=0$ 且 $\theta_{\mathrm{TST}}^{n,m}(l)\neq 0$；4) $x_{\mathrm{TST}}^{n,m}(l)=1$ 且 $\theta_{\mathrm{SAT}}^{n,m}(l)\neq 0$。

每个智能体在学习过程中都会将所有智能体考虑在内，多个智能体可能采取的动作相互排列组合，形成一个联合动作对集合。每个联合动作对所对应的 Q 值记录在 Q 表中，Q 值越高，对应的联合动作对被选中的概率越大。然而，大量的联合动作对受到问题式(21)中计算和频谱资源约束的限制，即在某个设备（卫星、TST 或 UE）处，所有任务所需的计算或频谱资源之和不应超过该设备拥有的总资源。为了避免各个智能体选中该类联合动作对，与它们相应的 Q 值被初始化为$-\infty$。

2.3.2　ε 贪心探索策略

在每次迭代的开始，每个智能体 l 根据ε 贪心探索策略[25]选择动作，即每个智能体 l 有ε的概率从动作空间中随机选择一个动作，即 $\forall\boldsymbol{a}_l\in\boldsymbol{\mathcal{A}}_l$；有 $1-\varepsilon$ 的概率选择 Q 值最高的最佳动作 $\boldsymbol{a}_l^*$。

2.3.3　最佳动作选择

最佳动作 $\boldsymbol{a}_l^*$是智能体 l 在当前状态下，能使当前 Q 值最大化的动作。在后续迭代中，智能体 l 将其他智能体历史选择动作的频率分布作为当前动作选择的指导。获取最佳动作 $\boldsymbol{a}_l^*$的策略用$\pi_l(S)$表示，可以表示为

$$\boldsymbol{a}_l^*=\pi_l(S)=\arg\max_{a_l}\sum_{\boldsymbol{a}_{-l}}\frac{\Phi(S,\boldsymbol{a}_{-l})}{n(S)}Q_l(S,(\boldsymbol{a}_l,\boldsymbol{a}_{-l})) \tag{23}$$

其中，计数器 $\Phi(S, \boldsymbol{a}_{-l})$用于记录除 l 外其他智能体在状态 S 下选择动作 $\boldsymbol{a}_{-l}$的次数，$n(S)$是历史上访问到状态 S 的总次数，$Q(S,(\boldsymbol{a}_l,\boldsymbol{a}_{-l}))$是在状态 S 下采取联合动作对$(\boldsymbol{a}_l,\boldsymbol{a}_{-l})$所对应的 Q 值。

2.3.4 Q 表更新

在状态 S 下执行动作 $\boldsymbol{a}_l$ 之后，每个智能体 l 观察环境、获得奖励 $R(S, \boldsymbol{a}_l)$，并依照式(24)更新 Q 值

$$Q_l(S,(\boldsymbol{a}_l,\boldsymbol{a}_{-l})) = (1-\alpha)Q_l(S,(\boldsymbol{a}_l,\boldsymbol{a}_{-l})) + \alpha(R(S,\boldsymbol{a}_l) + \gamma V_l(S')) \tag{24}$$

其中，$V_l(S') = \max_{\boldsymbol{a}'_l} \sum_{\boldsymbol{a}'_{-l}} \frac{\Phi(S',\boldsymbol{a}'_{-l})}{n(S')} Q(S',(\boldsymbol{a}'_l,\boldsymbol{a}'_{-l}))$ 表示智能体 l 在新状态 S' 下基于其他智能体历史动作选择频率分布下的期望奖励值，$\alpha\in(0,1)$和$\gamma\in(0,1)$分别表示学习率和折扣参数。本文在 i 次迭代中采用一种动态学习率 $\alpha^{(k)} = 0.95^k \alpha_0$，其中初始学习率$\alpha_0$为经验值。学习率是网络收敛的关键，将在第 4 节中通过仿真结果进行详细分析。

2.3.5 算法描述

基于 MARL 的 SHRCG 算法如算法 1 所示。每个智能体 l 分别从状态空间$\boldsymbol{\mathcal{S}}$和动作空间$\boldsymbol{\mathcal{A}}$中随机选择一个状态和一个动作，并初始化联合动作对集合、Q 表和策略π_l。在每次迭代中，智能体 l 有 $1-\varepsilon$ 的概率根据策略$\pi_l(S)$选择动作 $\boldsymbol{a}_l$，否则，为了探索，将会在动作空间$\boldsymbol{\mathcal{A}}_l$中随机选择一个动作 $\boldsymbol{a}_l$。在当前环境中执行动作 $\boldsymbol{a}_l$之后，智能体 l 获得奖励，根据式(23)和式(24)更新 Q 表和策略$\pi_l(S)$，以供下一次迭代使用。当达到最大迭代次数$i_{\max}$ 且满足系统时延需求（即β=1）时，停止迭代。

算法 1 基于 MARL 的 SHRCG 算法

输入 学习率 $\alpha^{(i)} \in (0,1]$；探索率 $\varepsilon > 0$

输出 任务卸载策略 x^*，资源分配方案 θ^*，$\boldsymbol{\phi}^*$

1) 初始化 $Q_l(S,(\boldsymbol{a}_l,\boldsymbol{a}_{-l}))$，$\forall S \in \prod_i^N \mathcal{S}$，$\boldsymbol{a}_l \in \boldsymbol{\mathcal{A}}_l$，$\boldsymbol{a}'_{-l} \in \prod_{r\neq l}^N \boldsymbol{\mathcal{A}}_r$；

2) for 每次迭代 $i < i_{\max}$ 或β= 0，智能体 l do；

3) $1-\varepsilon^{(i)}$的概率下，根据策略$\pi_l(S)$选择动作 $\boldsymbol{a}_l$，或在$\varepsilon^{(i)}$概率下，随机选择一个行动以探索；

4) 执行动作 $\boldsymbol{a}_l$；

5) 观察奖励 R；

6) 根据式(24)更新 Q 表；

7) 在状态 S 下，根据式(23)更新策略$\pi_l(S)$；

8) $i \leftarrow i+1$。

9) end for

2.4 天地异构资源协同博弈算法特性分析

本节对所提 SHRCG 算法特性进行了理论分析，包括动作空间缩减分析、联合动作对集合缩减分析、算法复杂度分析以及交互信息分析。

2.4.1 动作空间缩减分析

在基于 MARL 的 SHRCG 中，动作空间被构造为离散集，具体包括将资源分配方案$\boldsymbol{\theta}$ (l)和$\boldsymbol{\phi}$ (l)离散化成 c 个不同等级，将任务卸载策略 $\boldsymbol{x}(l)$离散化成 d 个不同等级。在缩减前，当每个智能体发布的任务由 MN 个 UE 参与时，每个任务包含 MN 个子任务，每个智能体需要对每个子任务中的 5 个资源分配变量和 2 个任务卸载变量进行决策。即 3 层设备上计算资源分配 $\boldsymbol{\theta}$ (l) =$\{ \theta_{\mathrm{UE}}^{n,m}(l)$， $\theta_{\mathrm{TST}}^{n,m}(l)$，$\theta_{\mathrm{SAT}}^{n,m}(l) \mid \forall m,n \}$和无线传输资源分配 $\boldsymbol{\phi}$ (l) = $\{ \phi_{\mathrm{UE}}^{n,m}(l)$，$\phi_{\mathrm{TST}}^{n,m}(l) \mid \forall m,n \}$，以及 UE 和 TST 上的任务卸载比例 $\boldsymbol{x}(l) = \{ x_{\mathrm{UE}}^{n,m}(l)$，$x_{\mathrm{TST}}^{n,m}(l) \mid \forall m,n \}$。简洁起见，本文假设 L 个智能体动作空间尺寸相同，可表示为

$$|\boldsymbol{\mathcal{A}}_1| = |\boldsymbol{\mathcal{A}}_2| = \cdots = |\boldsymbol{\mathcal{A}}_L| = |\boldsymbol{\mathcal{A}}| = c^{5MN} d^{2MN} \tag{25}$$

定理 1 当每个任务由 $M\times N$ 个 UE 产生原始数据时，缩减后的动作空间尺寸可表示为

$$|\boldsymbol{\mathcal{A}}^-| = \left(\frac{c^2d^2 - 4c^2d + 4c^2 + 3cd - 5c + 2}{c^2d^2} \right)^{MN} |\boldsymbol{\mathcal{A}}| \tag{26}$$

证明 详见附录 1。

2.4.2 联合动作对集合缩减

每个智能体在学习过程中都会将所有智能体考虑在内，多个智能体可能采取的行动相互排列组合，形成一个联合动作对集合。该联合动作对集合可以表示为

$$|\boldsymbol{\mathcal{Q}}| = |\boldsymbol{\mathcal{A}}_1\boldsymbol{\mathcal{A}}_2\cdots\boldsymbol{\mathcal{A}}_L| = |\boldsymbol{\mathcal{A}}_1||\boldsymbol{\mathcal{A}}_2|...|\boldsymbol{\mathcal{A}}_L| = |\boldsymbol{\mathcal{A}}|^L \tag{27}$$

然而，大量的联合动作对违反了式(21)中的计算和传输资源约束，这些联合动作对可以在 Q 表更新期间跳过。

定理 2 对于具有多颗卫星的天地算力网络，不参与 Q 表更新的联合动作对总数可以由式(28)计算而来。

$$\left|\mathcal{Q}^{-}\right|=\left[c^{L}-\frac{\prod_{0\leqslant i\leqslant L-1}(c+i)}{L!}\right]\left(\frac{|\mathcal{A}|}{c}\right)^{L}\cdot\frac{1-\left[\frac{\prod_{0\leqslant i\leqslant L-1}(c+i)}{L!c^{L}}\right]^{5MN}}{1-\frac{\prod_{0\leqslant i\leqslant L-1}(c+i)}{L!c^{L}}} \tag{28}$$

证明　详见附录 2。

2.4.3　算法复杂度分析

如式(23)和式(24)所示，在策略更新和 Q 表更新中，每个智能体将其他智能体的历史动作选择频率分布作为指导，SHRCG 算法每次迭代的计算复杂度为$\mathcal{O}(L|\mathcal{A}|^{L})$[26]，可通过减少动作空间大小$|\mathcal{A}|$和联合动作对集合的大小$|\mathcal{A}|^{L}$来降低计算复杂度。

2.4.4　信息交互分析

在天地算力网络中，为了获得最优方案，每个智能体都需要获取其他智能体的信息。然而，在现实中，在这些相互独立的智能体之间交换和共享信息是非常困难的。SHRCG 算法中，不同智能体之间需要交互的信息仅包括每个 UE 的数据产生速率及历史选择动作。每次迭代所交换的信息量仅为 $L(L-1)(MN+1)$。

3　实验结果与分析

本文基于系统时延、资源竞争描述、可拓展性、收敛速度和复杂度 5 个指标，与下列已有方案进行比较，评估网络性能。

1) 本地计算。所有数据在 UE 本地进行处理，而后将数据上传到相应的卫星。

2) 星端计算。产生的数据流直接上传到卫星，所有任务在卫星集中处理。

3) 中心式算法。任务被分成多个部分，分别在不同的层中处理，最后在卫星处汇聚。与本文所提机制不同，中心式算法由一个具有全局信息的中心控制器来决定任务卸载和资源分配策略，以提供一个最优方案。但是，现实生活中，卫星分属不同公司，上层并不存在中心控制器，因此本文算法仅在理论上可行，提供理论上的性能最优参考值。

3.1　参数设置

本文采用 Intel i7 处理器进行了仿真实验，主频 3.6 GHz，其上搭载 MATLAB 仿真软件。根据强化学习[27-28]和现有工作[29-30]的常用值，进行了仿真参数的设置，如表 2 所示。探索率为迭代次数的指数函数，表示为 $\varepsilon^{(k)}=0.95^{k}\varepsilon_{0}$，该探索率满足 $\lim_{k\to\infty}\varepsilon^{(k)}=0$。

表 2　　仿真参数设置

参数	值
每 UE CPU 频率/(cycle·s^{-1})	3×10^{1}
每 TST CPU 频率/(cycle·s^{-1})	8×10^{10}
每卫星 CPU 频率/(cycle·s^{-1})	2.4×10^{10}
处理每比特数据所需 CPU 周期/(cycle·bit^{-1})	100
TST-UE 链路带宽/MHz	20
TST-UE 链路子信道数量	500
TST-UE 链路发射功率/dBm	43
TST-UE 链路发射天线增益/dBi	31.2
TST-UE 链路噪声/(dBm·Hz^{-1})	−120
卫星-TST 链路带宽/MHz	100
卫星-TST 链路子信道数量	16
卫星-TST 链路发射功率/W	2
卫星-TST 链路发射天线增益/dBi	37.1
卫星-TST 链路噪声/(dBm·Hz^{-1})	−174
数据压缩率	10%
折扣率	0.9
初始学习率α_0	0.15
初始探索率ε_0	0.1

3.2　系统时延评估与分析

本文所提 SHRCG 算法在天地算力网络中的性能如图 4 所示（为了便于表示，图 4 纵坐标采用指数分布的表现形式），该网络包括 4 颗卫星、2 个 TST 和 2 个 UE，将资源分配和任务卸载策略分别离散化为 5 个和 6 个等级（即 c=5，d=6）。其中一个 UE 和与其相连的 TST 为卫星 1 和卫星 2 提供原始数据并承担部分任务计算，另一个 UE 和与其相连的 TST 为卫星 3 和卫星 4 提供服务。仿真结果表明，所有方案的系统时延均随数据产生速率的增长而增长。在不同的数据产生速率下，SHRCG 算法系统时延始终低于星端计算和本地计算，且可以逼近中心式算法。

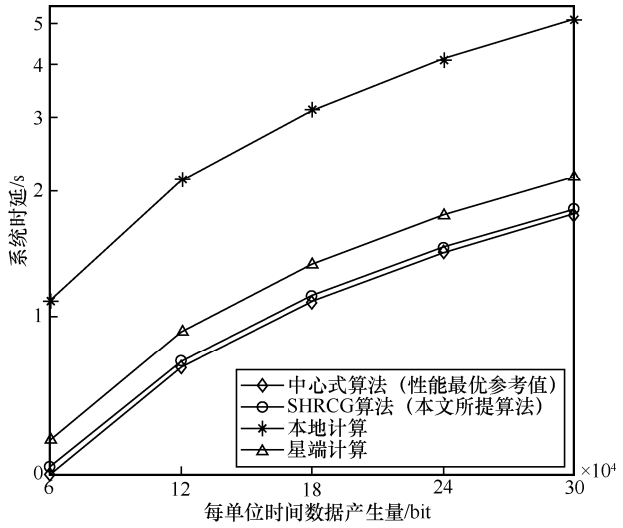


图 4　SHRCG 算法在天地算力网络中的性能

3.3　多卫星资源协作博弈评估与分析

由于任务分布式地部署在多个独立的卫星上，不同任务将竞争 TST 和 UE 上的计算和传输资源，以最小化自身时延。本节将基于 MARL 的 SHRCG 算法和基于单智能体强化学习的算法进行比较，以证明所提算法在天地异构资源协作博弈中的有效性。在单智能体强化学习中，将其中一个任务决策者作为智能体（称为智能体任务），通过学习确定任务卸载与资源分配策略；其余任务作为环境的一部分（称为环境任务），在剩余资源的前提下采取最优任务卸载与资源分配策略，从而实现任务处理。

单智能体强化学习和多智能体强化学习下任务时延和系统时延与迭代次数的关系分别如图 5 和图 6 所示（为了便于表示，图 6 纵坐标采用指数分布的表现形式）。考虑一个拥有 4 颗卫星的天地算力网络，资源分配方案和任务卸载策略分别被离散化为 5 个和 6 个等级。为简明地表现不同卫星的任务时延随着迭代次数的变化趋势，在 3.2 节仅将一颗智能体卫星及其中一颗环境卫星的仿真数据进行了绘制，另外 2 颗环境卫星不再展示，以避免多条曲线混杂造成阅读困难。在 3.3 节和 3.4 节，将进一步评估更多卫星场景下所提算法的有效性。

在单智能体强化学习过程中，智能体卫星倾向于尽可能多地占用 TST 和 UE 上的资源，直到环境卫星无资源可用。与智能体卫星相比，环境卫星的时延随迭代次数的增加而显著增加。数次迭代后，智能体卫星将占用 TST 和 UE 层的所有资源，此时，环境任务的时延趋向于无穷。综上，采用单智能体强化学习算法时，不同任务所占用的计算/传输资源存在巨大不平衡，导致系统时延趋向于无穷，系统不可用。

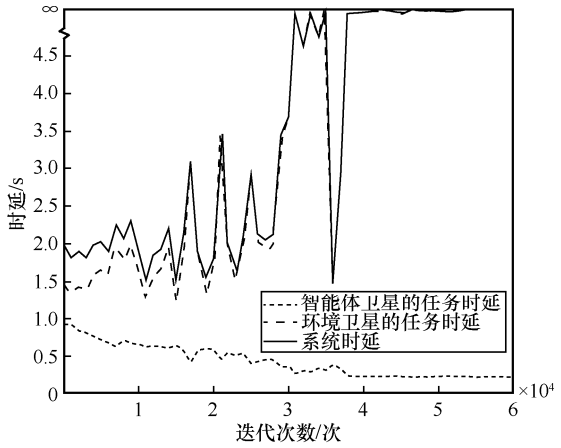


图 5　单智能体强化学习下任务时延和系统时延与迭代次数的关系

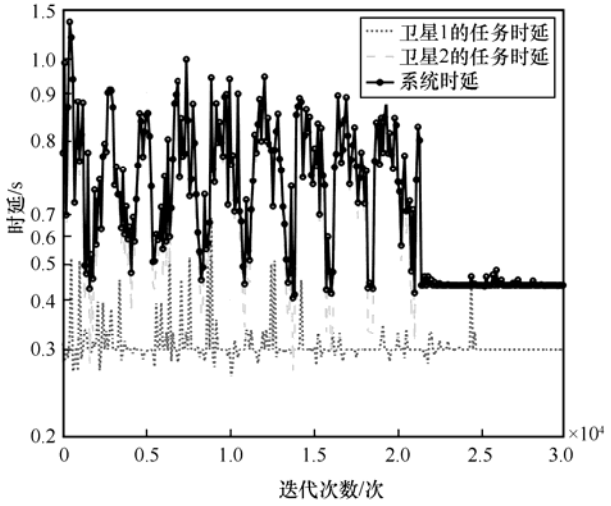


图 6　多智能体强化学习下任务时延和系统时延与迭代次数的关系

与单智能体强化学习算法相比，本文所提 SHRCG 算法中的所有任务都实现了较低的任务时延，从而最小化系统时延。对比图 5 和图 6 易知，SHRCG 算法中，每颗卫星进行决策时，对其他任务可能选择的决策进行了分析，从而可以更好地刻画多颗卫星间的博弈，进而实现系统时延最小化。

3.4　可拓展性分析

天地算力网络中系统时延与卫星数量的关系如表 3 所示。随着卫星数量的增加，TST 和 UE 需要利用其有限的计算和传输资源来处理更多的任务，因此系统时延呈线性增加。当系统中只存在一颗卫星时，SHRCG 算法的系统时延较低。随着网络中的卫星数目逐渐增加，系统时延也随之增

加。如图 4 所示，采用所提 SHRCG 算法，天地算力系统时延可以逼近中心式算法。

表 3　天地算力网络中系统时延与卫星数量的关系

卫星数量/颗	中心式算法/s	SHRCG/s	卫星计算/s	本地计算/s
1	1.171 8	1.178 5	1.210 0	1.502 6
2	1.336 4	1.339 3	1.410 0	2.004 6
3	1.492 6	1.503 5	1.610 0	2.506 6

天地算力网络中系统时延与 TST 数量的关系如表 4 所示。与表 3 中的情况不同，不同 TST 数量下的系统时延基本相同。这是因为当 TST 的数量变化时，UE 数量也随之增加。每个 UE 上的数据产生速率不变，每个设备上待处理的数据量保持不变，因此系统时延保持恒定。

表 4　天地算力网络中系统时延与 TST 数量的关系

TST 数量/个	中心式算法/s	SHRCG/s	卫星计算/s	本地计算/s
1	1.436 4	1.439 3	1.510 0	2.104 6
2	1.443 6	1.451 0	1.520 0	2.105 2
3	1.450 7	1.459 3	1.530 0	2.105 8

天地算力网络中系统时延与每个 TST 下的 UE 数量的关系如表 5 所示。随着与每个 TST 相连的 UE 数量的增加，系统时延也随之增加。随着 UE 增多，系统中的待处理数据随之增多，TST 和卫星的计算资源以及传输资源变得越来越稀缺。本地计算方案在 UE 本地处理整个任务，不受 TST 和卫星的计算资源的限制。此外，由于处理结果通常比原始数据小得多，本地计算方案可以大大减少待传输的数据量。因此，相较于其他方案，随着与每个 TST 相连的 UE 数量的增加，本地计算方案的系统时延变化相对较小。

表 5　天地算力网络中系统时延与每个 TST 下的 UE 数量的关系

UE 数量/个	中心式算法/s	SHRCG/s	卫星计算/s	本地计算/s
1	1.336 4	1.339 3	1.410 0	2.004 6
2	1.554 6	1.574 4	1.820 0	2.009 2
3	1.663 9	1.669 6	2.230 0	2.013 8

3.5　收敛性评估和分析

天地算力网络中卫星、TST、UE 的数量对算法收敛所需迭代次数如表 6 所示。随着卫星数量的增加，算法收敛速度减慢。由于 Q 表的大小是卫星数量的指数函数，因此收敛所需的迭代次数随卫星数量呈指数增长。当天地算力网络中只有一颗卫星时，学习曲线收敛到稳定值的迭代次数为 3.1×10^4。在双卫星的情况下，需要 4.9×10^4 次迭代才能收敛。含有 3 颗卫星的天地算力网络在 5.3×10^5 迭代之后收敛。相对而言，TST 数量及与每个 TST 连接的 UE 数量与 Q 表的大小之间呈现线性关系，TST 的数量及与每个 TST 连接的 UE 数量都不会显著影响收敛速度。

表 6　不同设备数量下算法收敛所需迭代次数

设备数量	卫星/次	TST/次	UE/次
1	3.1×10^4	4.9×10^4	4.9×10^4
2	4.9×10^4	5.3×10^4	5.8×10^4
3	5.3×10^5	6.1×10^4	5.5×10^4

3.6　算法复杂度评估与分析

如 2.3.3 节所示，每次迭代的计算复杂度为 $\mathcal{O}(L|\mathcal{A}|^L)$[26]，可通过减少动作空间大小$|\mathcal{A}|$显著降低复杂度。因此$|\mathcal{A}|$是度量复杂度的重要指标。如图 7 和图 8 所示，资源分配方案和任务卸载策略分别被离散化为 c 个和 d 个等级。由图 7 和图 8 可知，动作离散化的等级数越高，动作空间越大，计算复杂度越高，系统时延的计算精度越高。因此，可通过选择最优的动作离散化等级，实现计算复杂度和系统时延优化精度之间的权衡。

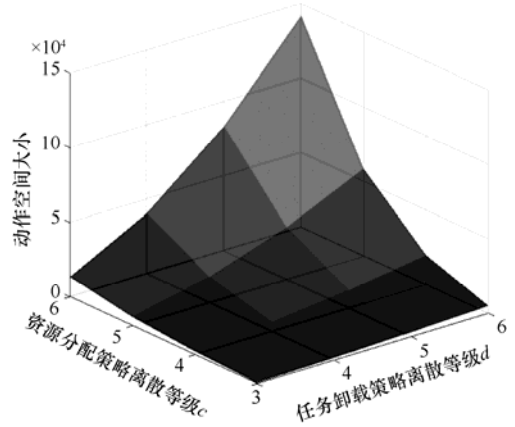


图 7　动作空间离散等级对动作空间大小的影响

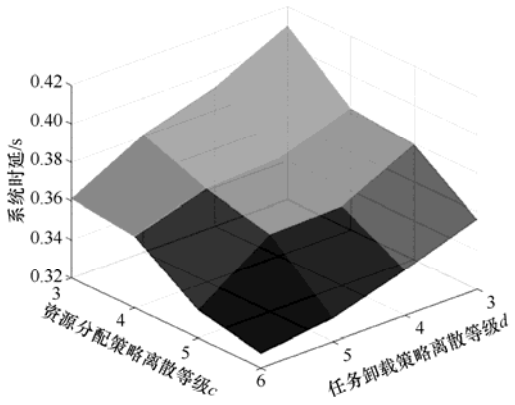


图 8　动作空间离散等级对系统时延的影响

4 结束语

为满足海量的高速数据服务需求，本文考虑了一种多卫星天地算力网络，利用网络中星地节点的算力等资源进行数据压缩，进而缓解天地一体化网络中的传输压力，降低系统时延。由于卫星分属于不同系列，彼此之间互相独立且存在特性差异，任务被分布式部署在多颗不同的卫星上。针对任务间计算域及频谱域资源竞争，提出了一种基于多智能体强化学习的天地异构资源协同博弈算法，以最小化系统时延。仿真结果表明：1) 与现有方案相比，所提算法能够充分利用网络中的计算与频谱资源，协调任务间资源竞争，从而实现系统时延最小化；2) 可通过优化马尔可夫决策过程中动作的离散化等级，实现算法复杂度和系统时延之间的权衡。

附录 1 定理 1 证明

当只有一个 UE 参与任务 l 时，根据动作空间缩减原则，表 7 中的动作将被从动作空间中删除。值得注意的是，这些原则是部分重叠的，重叠部分如表 8 所示。

表 7 动作空间缩减

缩减原则	占 $\lvert\mathcal{A}\rvert$ 的比例
$x_{\mathrm{UE}}=1$ 且 $x_{\mathrm{TST}}\neq 0$	$\frac{1}{d}\left(1-\frac{1}{d}\right)$
$x_{\mathrm{UE}}=0$ 且 $\theta_{\mathrm{UE}}\neq 0$	$\frac{1}{d}\left(1-\frac{1}{c}\right)$
$x_{\mathrm{TST}}=0$ 且 $\theta_{\mathrm{TST}}\neq 0$	$\frac{1}{d}\left(1-\frac{1}{c}\right)$
$x_{\mathrm{TST}}=1$ 且 $\theta_{\mathrm{SAT}}\neq 0$	$\frac{1}{d}\left(1-\frac{1}{c}\right)$

表 8 缩减原则重叠部分

重叠部分	占 $\lvert\mathcal{A}\rvert$ 的比例
$x_{\mathrm{UE}}=1$，$x_{\mathrm{TST}}=1$ 且 $\theta_{\mathrm{SAT}}\neq 0$	$\frac{1}{d}\frac{1}{d}\left(1-\frac{1}{c}\right)$
$x_{\mathrm{UE}}=0$，$x_{\mathrm{TST}}=0$，$\theta_{\mathrm{UE}}\neq 0$ 且 $\theta_{\mathrm{TST}}\neq 0$	$\left[\frac{1}{d}\left(1-\frac{1}{c}\right)\right]^2$
$x_{\mathrm{UE}}=0$，$x_{\mathrm{TST}}=1$，$\theta_{\mathrm{UE}}=0$ 且 $\theta_{\mathrm{TST}}\neq 0$	$\left[\frac{1}{d}\left(1-\frac{1}{c}\right)\right]^2$

因此，对于只有一个 UE 参与任务的情况，动作空间缩减可以缩减为 $\varsigma\lvert\mathcal{A}\rvert$，其中

$$\varsigma=1-\left\{\frac{1}{d}\left(1-\frac{1}{d}\right)+\frac{3}{d}\left(1-\frac{1}{c}\right)-\frac{1}{d}\frac{1}{d}\left(1-\frac{1}{c}\right)-2\left[\frac{1}{d}\left(1-\frac{1}{c}\right)\right]^2\right\}=\frac{c^2d^2-4c^2d+4c^2+3cd-5c+2}{c^2d^2} \tag{29}$$

随着 UE 的数量的增长，动作空间呈指数型缩减。当有 MN 个 UE 为任务 l 时收集原始数据时，动作空间可缩减至 $\varsigma^{MN}\lvert\mathcal{A}\rvert$。

附录 2 定理 2 证明

对于一个含有 2 颗卫星的网络，假设在联合动作对集合缩减中，只考虑 UE m 处的计算资源约束，则位于矩阵下三角区的资源对将违反约束条件限制，当 Q 表更新时，含有这些资源对的联合动作对将被跳过。由于矩阵是对称的，且每个动作还包含有其他任务卸载策略和资源分配方案，每个 $\theta_{\mathrm{UE}}^{n,m}(1)-\theta_{\mathrm{UE}}^{n,m}(2)$ 资源对将涉及动作空间中的 $\left(\frac{\lvert\mathcal{A}\rvert}{c}\right)^2$ 个联合动作对。因此，$\left[c^2-\frac{c(c+1)}{2}\right]\left(\frac{\lvert\mathcal{A}\rvert}{c}\right)^2$ 个联合动作对将不参与 Q 表更新。

参考文献：

[1] DI B Y, ZHANG H L, SONG L Y, et al. Ultra-dense LEO: integrating terrestrial-satellite networks into 5G and beyond for data offloading[J]. IEEE Transactions on Wireless Communications, 2019, 18(1): 47-62.

[2] SU Y T, LIU Y Q, ZHOU Y Q, et al. Broadband LEO satellite communications: architectures and key technologies[J]. IEEE Wireless Communications, 2019, 26(2): 55-61.

[3] 胡馨元, 邓若琪, 邸博雅, 等. 可重构全息超表面辅助卫星通信关键技术[J]. 电信科学, 2022, 38(10): 46-56.
HU X Y, DENG R Q, DI B Y, et al. Key technologies of satellite communications aided by reconfigurable holographic surfaces[J]. Telecommunications Science, 2022, 38(10): 46-56.

[4] VOIGT S, GIULIO-TONOLO F, LYONS J, et al. Global trends in satellite-based emergency mapping[J]. Science, 2016, 353(6296): 247-252.

[5] 王鹏飞, 邸博雅, 唐斌, 等. 面向广域海洋覆盖的密集低轨卫星星座[J]. 无线电通信技术, 2021, 47(4): 402-409.
WANG P F, DI B Y, TANG B, et al. Ultra-dense LEO satellite constellation design for wide ocean coverage area[J]. Radio Communications Technology, 2021, 47(4): 402-409.

[6] ZHOU Y, LIU L, WANG L, et al. Service-aware 6G: an intelligent and open network based on the convergence of communication, computing and caching[J]. Digital Communications and Networks, 2020, 6(3): 253-260.

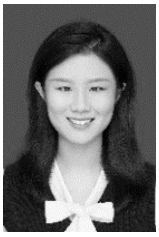
[7] ZHOU Y Q, TIAN L, LIU L, et al. Fog computing enabled future mobile communication networks: a convergence of communication and computing[J]. IEEE Communications Magazine, 2019, 57(5): 20-27.

[8] LIAO X L, HU X, LIU Z J, et al. Distributed intelligence: a verification for multi-agent DRL-based multibeam satellite resource allocation[J]. IEEE Communications Letters, 2020, 24(12): 2785-2789.

[9] LIU R, GUO K F, AN K, et al. Resource allocation for NOMA-enabled cognitive satellite-UAV-terrestrial networks with imperfect CSI[J].

IEEE Transactions on Cognitive Communications and Networking, 2023, 9(4): 963-976.
[10] WANG W L, WEI J, ZHAO S H, et al. Energy efficiency resource allocation based on spectrum-power tradeoff in distributed satellite cluster network[J]. Wireless Networks, 2020, 26(6): 4389-4402.
[11] ZHANG Y D, YIN L G, JIANG C X, et al. Joint beamforming design and resource allocation for terrestrial-satellite cooperation system[J]. IEEE Transactions on Communications, 2019, 68(2): 778-791.
[12] ZHANG S H, CUI G F, LONG Y T, et al. Joint computing and communication resource allocation for satellite communication networks with edge computing[J]. China Communications, 2021, 18(7): 236-252.
[13] FU S, GAO J, ZHAO L. Collaborative multi-resource allocation in terrestrial-satellite network towards 6G[J]. IEEE Transactions on Wireless Communications, 2021, 20(11): 7057-7071.
[14] DING C F, WANG J B, CHENG M, et al. Dynamic transmission and computation resource optimization for dense LEO satellite assisted mobile-edge computing[J]. IEEE Transactions on Communications, 2023, 71(5): 3087-3102.
[15] 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.
[16] 东方星. 我国高分卫星与应用简析[J]. 卫星应用, 2015(3): 44-48.
DONGFANG X. Analysis of high score satellite and its application in China[J]. Satellite Application, 2015(3): 44-48.
[17] 白徐祥, 官山林. 动中通卫星通信天线[J]. 无线通信技术, 2004, 13(1): 26-28, 32.
BAI X X, GUAN S L. Antenna design for satellite communication in motion[J]. Wireless Communication Technology, 2004, 13(1): 26-28, 32.
[18] GAO X Q, LIU R K, KAUSHIK A, et al. Dynamic resource allocation for virtual network function placement in satellite edge clouds[J]. IEEE Transactions on Network Science and Engineering, 2022, 9(4): 2252-2265.
[19] 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.
[20] ZHANG Y T, DI B Y, ZHENG Z J, et al. Distributed multi-cloud multi-access edge computing by multi-agent reinforcement learning[J]. IEEE Transactions on Wireless Communications, 2021, 20(4): 2565-2578.
[21] ZHAO N, YE Z Y, PEI Y Y, et al. Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2022, 21(9): 6949-6960.
[22] XU J, CHEN L X, REN S L. Online learning for offloading and autoscaling in energy harvesting mobile edge computing[J]. IEEE Transactions on Cognitive Communications and Networking, 2017, 3(3): 361-373.
[23] CLAUS C, BOUTILIER C. The dynamics of reinforcement learning in cooperative multiagent systems[C]//Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. New York: ACM Press, 1998: 746-752.
[24] TEYMOORI P, BOUKERCHE A. Dynamic multi-user computation offloading for mobile edge computing using game theory and deep reinforcement learning[C]//Proceedings of IEEE International Conference on Communications. Piscataway: IEEE Press, 2022: 1930-1935.
[25] GOMES E R, KOWALCZYK R. Dynamic analysis of multiagent Q-learning with ε-greedy exploration[C]//Proceedings of the 26th Annual International Conference on Machine Learning. New York: ACM Press, 2009: 369-376.
[26] SADHU A K, KONAR A. An efficient computing of correlated equilibrium for cooperative Q-learning-based multi-robot planning[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(8): 2779-2794.
[27] JU Y, CHEN Y C, CAO Z W, et al. Joint secure offloading and resource allocation for vehicular edge computing network: a multi-agent deep reinforcement learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(5): 5555-5569.
[28] WAQAR N, HASSAN S A, MAHMOOD A, et al. Computation offloading and resource allocation in MEC-enabled integrated aerial-terrestrial vehicular networks: a reinforcement learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 21478-21491.
[29] DENG R Q, DI B Y, ZHANG H L, et al. Ultra-dense LEO satellite constellations: how many LEO satellites do we need[J]. IEEE Transactions on Wireless Communications, 2021, 20(8): 4843-4857.
[30] WANG P F, DI B Y, SONG L Y. Mega-constellation design for integrated satellite-terrestrial networks for global seamless connectivity[J]. IEEE Wireless Communications Letters, 2022, 11(8): 1669-1673.

[作者简介]



张雨童（1995− ），女，山东德州人，北京大学博士生，主要研究方向为可重构智能超表面码本设计等。

彭煜明（1999− ），男，湖南株洲人，北京大学博士生，主要研究方向为超材料传感器设计等。

邸博雅（1992− ），女，黑龙江大庆人，博士，北京大学助理教授，主要研究方向为无线通信、边缘计算、车载网络、智能反射面和非正交多址接入等。

宋令阳（1979− ），男，辽宁抚顺人，博士，北京大学教授，主要研究方向为无线通信和网络、MIMO、OFDMA 以及信号处理和机器学习等。