

空天地一体化算力网络资源调度机制

王克文^{1,2}, 张维庭^{3*}, 孙童³

(1. 北京交通大学电气工程学院, 北京 100044; 2. 国能新朔铁路有限责任公司, 内蒙古 鄂尔多斯 010300;

3. 北京交通大学电子信息工程学院, 北京 100044)

摘要: 为满足卫星数据处理、车辆远程控制等快速响应和大范围覆盖的应用场景需求, 聚焦于采用分层控制和人工智能技术的方法, 设计一种空天地一体化算力网络资源调度机制。将空天地网络划分为 3 个域, 分别部署域控制器, 负责本地域的资源管理, 同时通过卫星和无人机的覆盖范围进行地面区域划分, 确保地面区域能够得到有效的服务保障, 以实现高效的数据传输和任务处理。为了优化空天地算力网络资源利用率, 引入多智能体强化学习算法, 对不同场景下产生的计算任务进行实时处理, 将每个域控制器视为具备任务调度和资源分配能力的智能体, 在满足时延和能耗的约束下, 通过协同学习和分布式决策实现计算任务智能调度和高效分配。实验结果表明, 该机制能够有效提高资源利用率和缩短任务响应时间。

关键词: 空天地一体化网络; 算力网络; 任务调度; 资源分配; 多智能体强化学习

中图分类号: TN929

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0069471

Resource Scheduling Mechanism for Space-Air-Ground Integrated Computing Power Networks

WANG Kewen^{1,2}, ZHANG Weiting^{3*}, SUN Tong³

(1. School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China;

2. Guoneng Xinshuo Railway Co., Ltd., Ordos 010300, Inner Mongolia, China;

3. School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

【Abstract】 In response to the increasing demand for fast response and large-scale coverage in application scenarios such as satellite data processing and vehicle remote control, this study focuses on utilizing hierarchical control and artificial intelligence technology to design a resource scheduling mechanism for space-air-ground integrated computing power networks. Air, space, and ground networks are divided into three domains, and domain controllers are deployed for resource management in the corresponding local domain. The areas are divided based on the coverage of satellites and drones to ensure that they can achieve effective service guarantees, efficient data transmission, and task processing. A multi-agent reinforcement learning-based scheduling algorithm is proposed to optimize resource utilization in space-air-ground integrated computing power networks, considering each domain controller is treated as an agent with task scheduling and resource allocation capabilities. Intelligent resource scheduling and efficient resource allocation for computing tasks are realized through collaborative learning and distributed decision-making with satisfactory delay and energy consumption constraints. Computing tasks are generated in different scenarios and processed in real time. Simulation results show that the proposed mechanism can effectively improve resource utilization and shorten task response time.

【Key words】 Space-Air-Ground Integrated Network (SAGIN); computing power network; task scheduling; resource allocation; multi-agent reinforcement learning

0 引言

随着全球互联网的快速发展, 涌现出大量的新兴应用服务, 如工业自动化、智能交通系统等, 这些服务产生的海量数据需要传输和处理^[1-2]。6G 通信技术为这些应用提供了高可靠、低延迟的连接服务。然而, 在偏远山区, 地面的蜂窝网络无法实现全覆

盖, 尤其遇到极端天气时, 地面的基础通信设施(如基站)容易受到破坏, 无法提供稳定的通信服务^[3]。为此, 研究学者提出了空天地一体化网络(SAGIN)^[4]。SAGIN 将天基网络、空基网络与地面网络相结合, 主要包括地面算力中心、服务器、无人机和卫星等计算节点, 为全球的智能应用提供泛在的计算服务^[5]。

收稿日期: 2024-03-04 修回日期: 2024-04-17

基金项目: 国家自然科学基金(62201029, 62394321); 中国博士后科学基金(2022M710007, BX20220029)。

通信作者 E-mail: *wtzhang@bjtu.edu.cn

在 SAGIN 架构中,天基网络和空基网络节点的计算资源和链路带宽资源有限,而其在地基网络中部署了很多服务器和智能算力中心,算力资源相对丰富。在能源方面,不同轨道上卫星的能量存储不同,并且在空基网络中的无人机等设备也都面临着能源约束。同时,卫星、无人机以及移动用户等节点的不断运动使得 SAGIN 成为一个高度时变的网络,其通信链路、网络拓扑和负载不断变化^[6]。SAGIN 中相关服务的调用通常涉及多维空间的协同。例如,紧急灾害遥感需要遥感卫星和无人机的配合^[7]。此外,对于计算密集型应用,如高清视频的传输处理,需要大量算力资源进行图像渲染和视频解码;对时延敏感型应用,如车辆自动驾驶,车内部传感器的数据需要实时处理^[8]。上述问题使得在 SAGIN 场景下对计算任务的调度和资源分配变得复杂,因此,提出新的计算架构解决此问题显得尤为重要。

传统的优化算法(如启发式算法,普遍依赖于基于经验和直觉预设的规则与策略)难以适应高度动态多变的 SAGIN 环境,并且该算法容易陷入局部最优,不利于实现对 SAGIN 的全局优化^[9]。此外,对于传统的基于博弈论的优化算法,SAGIN 的众多异构节点间的博弈关系错综复杂会导致建模难度大幅提升。

与上述传统优化算法相比,深度强化学习技术利用深度神经网络作为策略函数的近似器能够有效处理高维、连续的状态和动作空间^[10]。智能体通过与动态多变的 SAGIN 环境持续交互,根据环境反馈的奖励信号,自适应地调整和优化决策,无须依赖预先设计的复杂规则和精确模型^[11]。这种自主学习范式为智能化的任务调度和资源分配提供了新的解决思路。

本文的贡献可概括如下:

1)介绍了 SAGIN 的一般应用场景,在其 3 层网络中分别部署天基、空基和地基控制器,利用深度强化学习对各个域控制器进行赋能,实现动态分层控制。

2)建立智能应用设备与 SAGIN 的无线通信模型,提出计算任务调度和资源分配的时延和能耗模型,将该场景下计算任务调度、计算和存储等资源分配问题建模为马尔可夫过程,设计基于多智能体近端策略优化(MAPPO)的任务调度和资源分配算法。

3)对该算法进行仿真实验,验证了该算法可缩短复杂应用场景下计算任务的响应时间和降低各设

备的能耗开销。

1 系统模型

本节重点介绍了 SAGIN 的应用场景,构建了处理用户任务的计算模型和通信模型,将最小化时延和能耗的综合开销作为优化目标,求解优化问题。

1.1 网络场景

本文设计的 SAGIN 的一般场景如图 1 所示(彩色效果见《计算机工程》官网 HTML 版,下同),其中,重点关注了 SAGIN 多维的资源管理和分配。该 SAGIN 架构旨在通过全球的无缝覆盖以及空、天、地 3 个域的协同计算,以支持具有不同需求的各种应用场景的计算服务。SAGIN 控制架构包括空基网络、天基网络和地基网络。

在地基网络中,本文重点研究 3 种可以处理计算任务的设备,包括物联网设备、有线网络设备和专用计算设备。由于覆盖范围有限,5G/6G 蜂窝网络可能无法覆盖偏远地区,因此这些地区广泛部署了物联网来处理小型计算任务,例如视频监控、图像和声音等传感数据的融合。在 SAGIN 架构的支持下,物联网设备可以实现全球可扩展连接,与其他各域协同执行计算任务,将一些计算密集型任务通过无人机、卫星或者中继转发到云服务器和地面网络的算力中心进行计算,提高网络的计算能力。

在地面的有线网络中,路由器、交换机、网关等网络设备构成了核心网络,主要负责逐跳的计算任务转发。计算设备通常位于网络边缘,主要集中在一些大型算力中心中,算力中心的服务器由一个或多个处理模块组成,包括中央处理单元(CPU)、图形处理单元(GPU)和现场可编程门阵列(FPGA)等,负责即时任务处理^[12]。通过路由器和交换机等网络设备,用户产生的计算任务可传输到计算设备中。另外,通过引入计算能力,路由器设备可以同时具备转发和计算功能。当计算任务经过计算赋能路由器设备时,它可以在本地处理任务,也可以将任务转发到其他设备^[13]。因此,对于一些小型计算任务,在任务转发沿途即可完成计算并返回给用户。

在空基网络中,飞行无人机可以作为边缘服务器,为地面用户提供低时延的边缘缓存和计算能力。NVIDIA Jetson 系列的轻型无人机嵌入了轻量级的人工智能平台,可以将推理阶段的深度学习任务调度到无人机层的轻量级 AI 平台。每架无人机配置有固定的飞行轨迹,以服务于某块特定的区域^[14]。

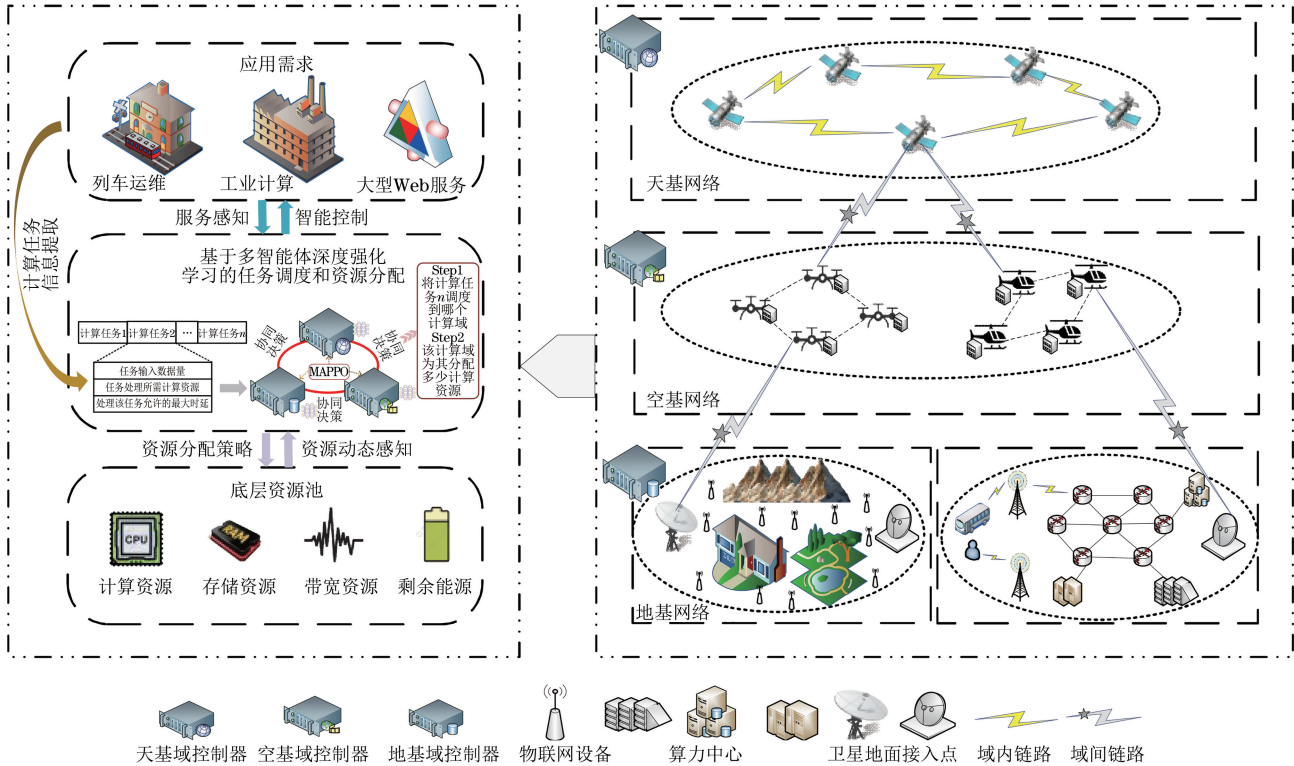


图 1 空天地一体化网络应用场景

Fig.1 Application scenarios of the SAGIN

在天基网络中,在一些用户密度低、可用网络资源有限(如部分地区只有天基网络覆盖)的地区,可通过与卫星通信满足体验质量(QoE)和服务质量(QoS)需求。同时,在卫星运行过程中,也会生成许多计算任务,例如地球观测数据处理、故障诊断和管理等^[15]。然而,由于近地球轨道(LEO)卫星有限的计算能力和星上能源供应能力,其一般执行轻量级的计算任务,而地球静止轨道(GEO)和中地球轨道(MEO)卫星比LEO卫星拥有更可观的计算资源和太阳能能源^[16]。因此,若GEO卫星计算资源充足,则可将MEO产生的计算任务调度到GEO处理。此外,对一些没有地面网络覆盖而且时间紧迫的服务,将任务直接调度到卫星上处理,可满足时延的需求。在本文中,考虑由多颗卫星实现某块区域全覆盖的情形。

本文将SAGIN架构划分为天基域、空基域和地基域,分别部署天基控制器、空基控制器和地基控制器,主要负责各自物理域中信息聚合、任务的调度和资源管理分配。对于智能应用设备产生的计算任务,域控制器将决定该任务的调度位置和资源分配量。在地基网络中,地基控制器可以部署在地面中继站或基站中。在空基网络中,空基控制器部署在无人机中。在天基网络中,天基控制器可以部署在LEO或者卫星地面站中。首先通过新型带内遥测

技术,灵活地感知3个域中的资源信息^[17],其中包括可用的计算、存储、带宽、频谱资源以及剩余能源供应,这些资源信息作为深度强化学习(DRL)模型的状态输入。将每个域控制器视为一个智能体,智能体将根据状态进行分析和学习,并做出相应的动作。然后,环境会给出该动作的奖励信号,以指导智能体调整其决策。不断重复上述过程,实现网络管理和优化。域控制器需要动态做出对计算任务的调度和资源分配决策,即:将计算任务调度到哪个计算域执行,对于本地计算域分配多少资源来执行该计算任务。

假设在该网络场景中,有 n 种发起计算任务的智能应用设备,并且每种设备最多同时发起一个计算任务。定义任务集合 $A = \{A_1, A_2, \dots, A_N\}$ 。对于每个计算任务 A_n ,用 $A_n = \{c_n, x_n, \tau_n\}$ 来描述, c_n 表示执行该任务所需要的计算资源, x_n 表示任务的输入数据量, τ_n 表示执行该任务允许的最大时延。

由于太空中的卫星的位置和天空中的无人机的位置会发生变化,只有在卫星和无人机与地面用户之间满足一定关系时,地面终端用户才能与其进行通信。文献[18]提出了卫星和无人机覆盖时间模型,将卫星覆盖时间定义为 M ,假设无人机的飞行周期和卫星覆盖时间相同。无人机飞行在固定高度

上,为了使无人机的位置相对用户近乎保持不变,将 M 分割为长度为 $\delta_t = M/T$ 的足够小的时隙,并且在不同时段下的无人机的位置发生变化,即在时段 t ,卫星和无人机的覆盖范围不变,SAGIN 的资源状态也是不变的,定义时段集合 $t \in \{1, 2, \dots, T\}$ 。

1.2 通信模型

本文所述的 SAGIN 架构中存在 3 种无线通信,即智能应用设备与地面、天空、太空的无线通信,接下来将分别介绍这 3 种通信模型。首先,智能应用设备 n 与地面的数据传输速率可以表示如下:

$$R_n^g = B_n^g \log \left(1 + \frac{p_n h_n^g (d_n^g)^{-\theta}}{\sigma_g^2} \right) \quad (1)$$

式中: B_n^g 表示终端设备与地面通信时所分配到的链路带宽资源; σ_g^2 表示信道的背景噪声; p_n 表示计算任务的传输功率; h_n^g 表示信道增益; d_n^g 表示终端设备到地面算力节点的距离; θ 表示距离衰减因子^[19]。

类似地,可以分别得出智能应用设备 n 与天空、太空的数据传输速率,分别表示如下:

$$R_n^a = B_n^a \log \left(1 + \frac{p_n h_n^a (d_n^a)^{-\theta}}{\sigma_a^2} \right) \quad (2)$$

$$R_n^p = B_n^p \log \left(1 + \frac{p_n h_n^p (d_n^p)^{-\theta}}{\sigma_p^2} \right) \quad (3)$$

式中: B_n^a 、 B_n^p 分别表示终端设备与天空、太空通信时所分配到的链路带宽资源。

1.3 时延与能耗模型

由于本文将最小化时延和能耗的综合开销作为优化目标,因此提出时延与能耗模型。不同节点的计算能力不同,执行计算任务时会产生不同的计算时延。当计算任务进行调度处理时,任务数据需要传输到目标节点,产生传输时延,传输时延受到信道质量、分配的带宽资源等因素的影响,数据包转发时的排队时延相对计算时延和传输时延较小,可忽略不计。对不同的任务调度和资源分配决策,不同设备产生的能耗也不相同,这些模型为后续算法的设计以及性能的评估提供了基础。

1.3.1 计算时延

计算时延即计算任务在算力设备上执行完毕所需要的时间。在本文所述的架构中,车辆的计算任务可以在地基网络、空基网络或天基网络上执行,则在 3 个域中的计算时延分别可以表示如下:

$$T_n^{g, \text{comput}} = \frac{c_n}{c_n^g} \quad (4)$$

$$T_n^{a, \text{comput}} = \frac{c_n}{c_n^a} \quad (5)$$

$$T_n^{p, \text{comput}} = \frac{c_n}{c_n^p} \quad (6)$$

式中: $T_n^{g, \text{comput}}$ 、 $T_n^{a, \text{comput}}$ 、 $T_n^{p, \text{comput}}$ 分别表示任务在地基网络、空基网络和天基网络中的计算时延; c_n^g 、 c_n^a 、 c_n^p 分别表示在地基网络、空基网络和天基网络中分配给计算任务的计算资源。

此外,当智能应用设备 n 的计算任务被调度到 SAGIN 架构的某个域执行时,必须为该任务分配足够的存储空间,否则即使为计算任务分配了足够的计算资源,任务仍旧会中断或失败。

1.3.2 传输时延

传输时延指计算任务从无线链路传输到指定计算域所需要的时间。相应地,计算任务被调度到地基网络、空基网络、天基网络的时间可由下式表示:

$$T_n^{g, \text{trans}} = \frac{x_n}{R_n^g} \quad (7)$$

$$T_n^{a, \text{trans}} = \frac{x_n}{R_n^a} \quad (8)$$

$$T_n^{p, \text{trans}} = \frac{x_n}{R_n^p} \quad (9)$$

综上所述,对于计算任务 A_n ,从产生到最终被执行完成返回的时延等于传输时延、计算时延和接收时延三者之和。一般而言,计算结果的数据量通常很小,因此可以忽略接收结果的时延。计算任务 A_n 的时延可以表示如下:

$$T_n^g = T_n^{g, \text{trans}} + T_n^{g, \text{comput}} \quad (10)$$

$$T_n^a = T_n^{a, \text{trans}} + T_n^{a, \text{comput}} \quad (11)$$

$$T_n^p = T_n^{p, \text{trans}} + T_n^{p, \text{comput}} \quad (12)$$

1.3.3 能耗

对于空天地 3 个域中的计算设备,处理每个计算任务都有一定的能量消耗。文献[18]提出了对设备能耗的具体表达式。以处理器为 CPU 举例,定义 κ 为 CPU 的有效电容系数(取决于其处理器芯片的结构), C 为 CPU 计算每比特所需要的周期数,则地基网络、天基网络、空基网络处理计算任务 A_n 的计算能耗分别表示如下:

$$E_n^{g, \text{comput}} = \kappa C (c_n^g)^2 \quad (13)$$

$$E_n^{a, \text{comput}} = \kappa C (c_n^a)^2 \quad (14)$$

$$E_n^{p, \text{comput}} = \kappa C (c_n^p)^2 \quad (15)$$

对于发起计算任务的终端设备,其能耗主要是由传输计算任务数据形成的,向地基网络、空基网络、天基网络传输计算任务时的能耗分别表示如下:

$$E_n^{g, \text{local}} = p_n T_n^{g, \text{trans}} \quad (16)$$

$$E_n^{a, \text{local}} = p_n T_n^{a, \text{trans}} \quad (17)$$

$$E_n^{p, \text{local}} = p_n T_n^{p, \text{trans}} \quad (18)$$

因此,计算任务被调度到 3 个域时所产生的系统能耗分别表示如下:

$$E_n^g = E_n^{g, \text{comput}} + E_n^{g, \text{local}} \quad (19)$$

$$E_n^a = E_n^{a, \text{comput}} + E_n^{a, \text{local}} \quad (20)$$

$$E_n^p = E_n^{p, \text{comput}} + E_n^{p, \text{local}} \quad (21)$$

1.4 优化问题描述

对于计算任务 A_n , 本文规定不可将其同时调度到多个域中执行, 因此, 在时刻 t , 对于任务调度决策, 定义调度位置 $w_{t,n}^i = \{0, 1\}, i \in \{g, a, p\}$, 式中, $w_{t,n}^i$ 是一个布尔量, 表示是否将计算任务 A_n 调度到第 i 个域, g, a, p 分别表示地基网络、空基网络、天基网络。

本文的目标是对智能应用设备计算任务的调度决策、任务传输时延和执行能耗进行联合优化, 以最小化总时延和能耗的综合开销的方式处理完成计算任务。

定义在时刻 t 的系统总成本:

$$C_t = \sum_{n=1}^N \sum_i (\beta w_n^i T_n^i + (1-\beta) w_n^i E_n^i) \quad (22)$$

式中: $i \in \{g, a, p\}$; T_n^i 表示计算任务 A_n 被调度到第 i 个域的传输时延和处理时延的和; E_n^i 表示计算任务 A_n 被调度到第 i 个域的传输能耗和计算能耗的和。

因此, 优化目标问题转化为最小化系统成本问题:

$$\min C_t$$

$$\text{s. t. C1: } \min(T_n^g, T_n^a, T_n^p) \leq \tau_n$$

$$\text{C2: } w_n^g + w_n^a + w_n^p = 1$$

$$\text{C3: } \sum_{n=1}^N E_n^{g, \text{comput}} \leq e_g$$

$$\text{C4: } \sum_{n=1}^N E_n^{a, \text{comput}} \leq e_a$$

$$\text{C5: } \sum_{n=1}^N E_n^{p, \text{comput}} \leq e_p \quad (23)$$

式中: e_g, e_a, e_p 分别表示地基网络、空基网络、天基网络的可用能源。

2 基于 MAPPO 的解决方案

在系统建模部分, 天基、空基、地基网络分别部署了相应的域控制器, 它们需要协同工作以实现全局最优的任务调度和资源分配。MAPPO 算法用于处理多个智能体的协同决策问题, 将每个域控制器视为一个智能体, 仅需观测和处理本域内的资源信息和环境状态, 作出决策并执行相应动作, 这种分布式执行模式降低了单个域控制器的计算和存储负担, 提高了模型训练时的计算效率。此外, 通过利用策略共享和联合学习机制, 各智能体之间可以实现网络参数的同步和全局视角的获取。

综上所述, 本文提出了一种基于 MAPPO 的任务调度和资源分配策略, 以实现任务响应时间和设备能耗的优化, 架构如图 2 所示。

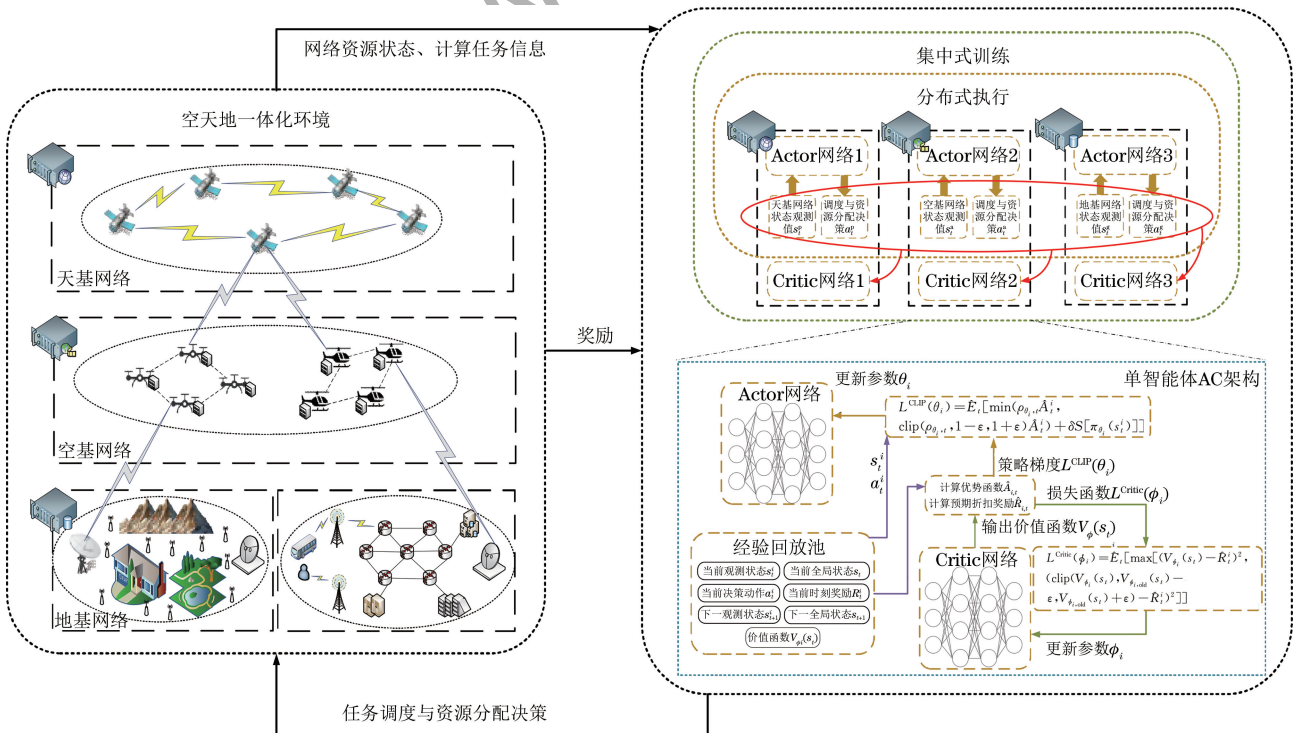


图 2 基于 MAPPO 的任务调度和资源分配架构

Fig.2 Architecture of task scheduling and resource allocation based on MAPPO

2.1 马尔可夫决策过程建模

将任务调度和资源分配问题建模为典型的多智能体马尔可夫决策过程,用元组 $(\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R})$ 来表示,其中,每个域控制器是一个智能体, $\mathbf{S}$ 是所有智能体 $i \in \{g, a, p\}$ 共享的全局状态空间, $\mathbf{A}$ 表示所有智能体的联合动作空间, $a^i \in \mathbf{A}$ 表示智能体 i 的决策动作^[20], $\mathbf{P}$ 表示状态转移函数, $\mathbf{R}$ 是计算全局奖励的函数。在时刻 t , 每个智能体根据策略 $\pi_i(a^i | s_t^i)$ 选择一个决策动作 a^i , 所有智能体的共同目标是使得系统成本最低。

2.1.1 状态空间

当前时刻 t 的全局状态 s_t 可表示如下:

$$s_t = \{s_t^g, s_t^a, s_t^p\} \quad (24)$$

$$s_t^g = \{A_n, c_t^g, o_t^g, b_t^g, e_t^g\} \quad (25)$$

$$s_t^a = \{A_n, c_t^a, o_t^a, b_t^a, e_t^a\} \quad (26)$$

$$s_t^p = \{A_n, c_t^p, o_t^p, b_t^p, e_t^p\} \quad (27)$$

式中: s_t^g, s_t^a, s_t^p 分别代表地基、空基、天基网络中的资源状态和计算任务的相关信息,这是由每个域控制器感知得到的,包括计算资源、存储资源、带宽资源、可用能源。

具体地: A_n 表示在时刻 t 需要调度决策的计算任务的相关信息; c_t^g, c_t^a, c_t^p 分别代表地基、空基、天基网络的计算资源,即执行计算任务的能力; o_t^g, o_t^a, o_t^p 分别代表地基、空基、天基网络的存储资源,即设备用于存储计算任务数据的能力; b_t^g, b_t^a, b_t^p 分别代表地基、空基、天基网络的带宽资源,即计算设备用于与其他域或其他本地设备进行通信的能力,带宽资源决定了其在传输数据时的速率和容量; e_t^g, e_t^a, e_t^p 分别代表地基、空基、天基网络的可用能源,即计算设备的电量,在决策时该信息可以体现设备能否支持计算任务执行完成。

2.1.2 动作空间

多智能体的动作空间可表示如下:

$$\mathbf{A} = \{a^i\}_{i \in \{g, a, p\}} \quad (28)$$

在当前时刻 t , 定义智能体 i 对计算任务 A_n 的调度和资源分配决策集合:

$$a_t^i = \{w_{t,n}^i, d_{t,n}^i\} \quad (29)$$

$$L^{\text{Critic}}(\phi_i) = \hat{E}_t[\max[(V_{\phi_i}(s_t) - \hat{R}_t^i)^2, (\text{clip}(V_{\phi_i}(s_t), V_{\phi_{i,\text{old}}}(s_t) - \epsilon, V_{\phi_{i,\text{old}}}(s_t) + \epsilon) - \hat{R}_t^i)^2]] \quad (32)$$

Critic 网络通过最小化损失函数来不断更新参数 ϕ_i ^[23]。为了缓解参数过度更新, $L^{\text{Critic}}(\phi_i)$ 的计算也采用了对更新后值函数 $V_{\phi_i}(s_t)$ 进行裁剪的方法。 $\hat{R}_t^i$ 表示预期折扣奖励,计算公式如下:

$$\hat{R}_t^i = r_t^i + \gamma V_{\phi_{i,\text{old}}}(s_{t+1}) \quad (33)$$

具体地, $\forall n \in \{1, 2, \dots, N\}, \forall i \in \{g, a, p\}$, $w_{t,n}^i$ 是一个布尔量, $w_{t,n}^i = \{0, 1\}$, $w_{t,n}^i$ 是调度决策,表示能否将任务 A_n 调度到第 i 个域。

$$d_{t,n}^i = \{c_{t,n}^i, o_{t,n}^i, b_{t,n}^i\} \quad (30)$$

式中: $d_{t,n}^i$ 是资源分配决策; $c_{t,n}^i, o_{t,n}^i, b_{t,n}^i$ 分别表示给该计算任务分配的计算、存储、带宽资源。

2.1.3 奖励函数

本文的优化目标是 minimized 计算任务调度处理的时延和能耗,对每个域控制器,只考虑调度到本地域内的计算任务的时延和能耗最小。奖励函数的设计应当能够引导智能体进行自主学习,从而做出最小化计算任务处理的时延和能耗的动作。因此,奖励函数设置如下:

$$r_t^i = \frac{1}{C_t} \quad (31)$$

2.2 基于 MAPPO 的任务调度和资源分配算法设计

本节将详细阐述解决时延和能耗联合优化的 MAPPO 算法。MAPPO 算法基于多智能体上的 AC (Actor-Critic) 架构和近端策略优化 (PPO) 算法^[21]。

在本文的 MAPPO 算法中,每个域控制器被视为一个独立的智能体,以地基控制器为例,介绍如何使用 MAPPO 算法解决 SAGIN 场景下计算任务调度和资源分配问题,天基控制器和空基控制器的原理与之类似。

每个智能体包含两个网络,即参数为 θ_i 的 Actor 网络和参数为 ϕ_i 的 Critic 网络。在时刻 t , 地基控制器动态感知本地域的资源信息 s_t^g , 并将其和计算任务的相关信息输入 Actor 网络^[22]。Actor 网络生成动作概率分布 $\pi_{\theta_i}(a_t^g | s_t^g)$, 即:获取任务调度和资源分配策略。地基智能体执行决策 a_t^g 后,获得瞬时奖励 r_t^g 和下一个时刻的状态值 s_{t+1}^g , 并将 s_t^g, a_t^g, r_t^g 和 s_{t+1}^g 存储在经验回放池中。同时, Critic 网络根据在时刻 t 的所有智能体本地状态观测值的集合生成状态值函数 $V_{\phi_i}(s_t)$, 根据观测值 s_{t+1}^g 生成状态值函数 $V_{\phi_i}(s_{t+1})$ 。由此,得到 Critic 网络的损失函数:

根据文献[24-25]的工作,通过裁剪 $\rho_{\theta_i,t}$ 以获得新的策略优化目标函数,进而优化 Actor 网络,使得该算法可获得更好的性能。因此,通过最大化目标函数来更新 Actor 网络的参数 θ_i :

$$\hat{R}_t^i = r_t^i + \gamma V_{\phi_{i,\text{old}}}(s_{t+1}) \quad (34)$$

$$L^{\text{CLIP}}(\theta_i) = \hat{E}_t[\min(\rho_{\theta_i,t} \hat{A}_t^i, \text{clip}(\rho_{\theta_i,t}, 1-\epsilon, 1+\epsilon) \hat{A}_t^i) + \delta S[\pi_{\theta_i}(s_t^i)]] \quad (35)$$

式中: $\rho_{\theta_i,t} = \frac{\pi_{\theta_i}(a_t^i | s_t^i)}{\pi_{\theta_i, \text{old}}(a_t^i | s_t^i)}$ 表示更新后的策略参数 θ_i 与更新前的比值, 取值为 $[1-\epsilon, 1+\epsilon]$, 从而保证了更新范围; $S[\pi_{\theta_i}(s_t^i)]$ 表示在策略 $\pi_{\theta_i}(s_t^i)$ 下通过感知地基网络资源状态 s_t^i 而输出的任务调度和资源分配决策的熵, 熵的系数为 δ ; $\hat{A}_t^i$ 根据文献[26]使用泛化优势估计(GAE)计算。 $\hat{A}_t^i$ 计算公式如下:

$$\hat{A}_t^i = \delta_t + (\gamma\lambda)\delta_t + (\gamma\lambda)^2\delta_t + \dots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \quad (36)$$

$$\delta_t = r_t^i + \gamma V_{\phi_{\text{old}}}(s_{t+1}) - V_{\phi_{\text{old}}}(s_t) \quad (37)$$

式中: $V_{\phi_{\text{old}}}(s_t)$ 是在时刻 t 的状态价值函数; $\gamma\lambda$ 是折扣因子。

$L^{\text{CLIP}}(\theta_i)$ 反向传播到 Actor 网络以更新参数 θ_i 。

基于 MAPPO 的计算任务调度和资源分配算法的伪代码如算法 1 所示。

算法 1 基于 MAPPO 的计算任务调度和资源分配算法

输入 在初始时刻设备(用户)产生的计算任务的相关信息 $A_n = \{c_n, x_n, \tau_n\}$ 和学习率 l_r

输出 计算任务调度位置决策以及对计算任务分配的计算资源 c_n^i 、存储资源 o_n^i 、带宽资源 b_n^i

1. 初始化每个智能体的 Actor 网络的参数 θ_i 和 Critic 网络的参数 ϕ_i
2. 随机执行计算任务调度和资源分配决策 a_0^i , 计算初始局部状态 s_t^i 和全局状态 s_t ;
3. for $N_{\text{iteration}} = 1, 2, \dots, N_{\text{episode}}$ do
4. for $t = 1, 2, \dots, T$ do
5. for agent $i = 1, 2, 3$ do

6. 根据 Actor 网络获得决策动作 $a_t^i = \pi_{\theta_i}(s_t^i)$;
7. 根据 Critic 网络获得全局状态 s_t 的价值函数 $V_{\phi_i}(s_t)$;
8. end for
9. 计算每个域控制器的奖励 r_t^i 、下一个时刻该域控制器本地状态 s_{t+1}^i 和全局状态 s_{t+1} ;
10. 将 $[s_t^i, s_t, a_t^i, V_{\phi_i}(s_t), r_t^i, s_{t+1}^i, s_{t+1}]$ 放入经验回放池;
11. end for
12. 计算 $\hat{A}_t^i$ 和 $\hat{R}_t^i$;
13. for $N_{\text{training}} = 1, 2, \dots, N_{\text{epoch}}$ do
14. for agent $i = 1, 2, 3$ do
15. 从经验回放池中随机取样
16. 用梯度上升算法更新 Actor 网络 $\theta_i = \theta_i + \alpha \frac{\partial L^{\text{CLIP}}(\theta_i)}{\partial \theta_i}$;
17. 用梯度下降算法更新 Critic 网络 $\phi_i = \phi_i - \alpha \frac{\partial L^{\text{critic}}(\phi_i)}{\partial \phi_i}$;
18. end for
19. end for
20. end for

3 实验与结果分析

本文对所提出的基于 MAPPO 的计算任务调度和资源分配算法进行了仿真实验, 为了验证该算法的有效性, 将其与基于 D3QN (Dueling Double DQN) 的算法和随机执行策略进行了性能对比。在随机执行策略中, 计算任务的调度位置和资源分配决策都是随机的。在本文中, 验证算法性能的指标为系统奖励和平均时延, 其中系统奖励为系统成本的倒数。实验环境的参数设置如表 1 所示。

表 1 实验环境参数设置

Table 1 Experimental environment parameter setting

仿真参数	具体描述	取值
p_n /dBm	计算任务的传输功率	55
$\sigma_g^2, \sigma_a^2, \sigma_p^2$ /dBm	信道的背景噪声	65
h_n^g, h_n^a, h_n^p /dBm	信道增益	15
θ	距离衰减因子	3
c_g, c_a, c_p / (GHz·s ⁻¹)	地基、空基、天基网络节点计算资源容量	100, 70, 50
o_g, o_a, o_p /MB	地基、空基、天基网络节点存储资源容量	200, 160, 120
e_g, e_a, e_p /MB	地基、空基、天基网络节点能源容量	90, 40, 60
c_n /GHz	计算任务所需要的计算资源	服从在[1, 2]上的均匀分布
x_n /MB	计算任务输入数据量	服从在[1, 3]上的均匀分布
τ_n /s	执行该任务允许的最大时延	服从在[0.5, 3.0]上的均匀分布

不同算法下的系统奖励如图 3 所示。从图 3 中可以看出,基于 MAPPO 的任务调度和资源分配算法相比于其他两种算法效果更好。在收敛速度上,基于 MAPPO 的算法明显优于基于 D3QN 的算法,当迭代次数约为 40 时,基于 MAPPO 的算法开始收敛并趋于稳定;在系统奖励上,基于 MAPPO 的算法的系统奖励是最高的,在随机执行策略下,系统奖励是最低的,这是因为计算任务调度和资源分配决策是随机的,没有考虑到计算任务的资源需求和网络资源状态,所以在每次迭代时,动作决策选择多,时延和能耗无法随着迭代次数的增加而优化。

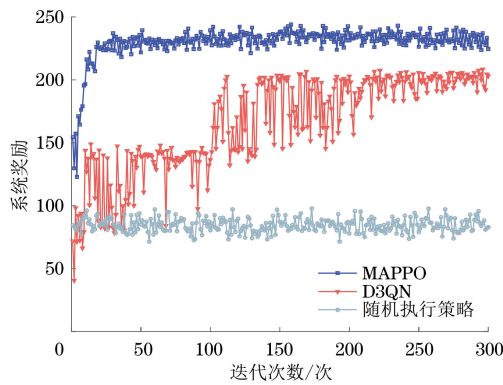


图 3 不同算法下的系统奖励

Fig.3 System rewards under different algorithms

图 4 展示了学习率分别为 5×10^{-2} 、 5×10^{-3} 、 5×10^{-4} 、 5×10^{-5} 时系统奖励随迭代次数增加的变化。当学习率为 5×10^{-3} 时,在迭代次数约为 40 时,算法开始趋于收敛,之后的收敛情况较为稳定;当学习率为 5×10^{-4} 和 5×10^{-5} 时,在迭代次数约为 50 时,算法开始趋于收敛,但是当学习率为 5×10^{-4} 时,在迭代次数约为 75 时算法收敛出现了不稳定情况;当学习率为 5×10^{-2} 时,迭代次数在 300 之内,算法收敛情况不稳定。因此,本文设置学习率为 5×10^{-3} 。

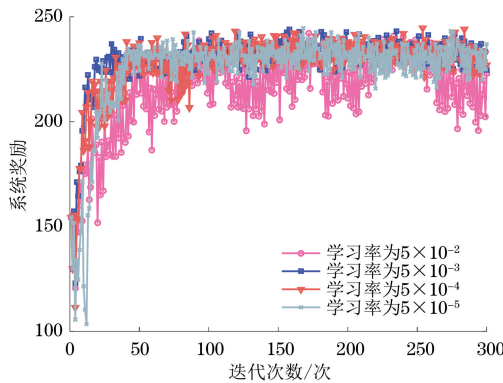


图 4 不同学习率下的系统奖励

Fig.4 System rewards under different learning rates

不同算法在不同计算资源下的系统奖励如图 5 所示。在本文实验环境中,初始设置地基、空基、天基网络的计算资源分别为 100、70、50 GHz,为了探讨网络中计算资源对系统奖励的影响,将实验环境资源分别更改为初始资源的 50%、60%、70%、80%、90%、100%,得到网络中不同计算资源下的系统奖励。从图 5 中可以发现,系统奖励随计算资源的增加而增加,在 3 种算法中,本文所提出的基于 MAPPO 的算法的系统奖励最高,当计算资源比例为 0.9 时,基于 MAPPO 的算法、基于 D3QN 的算法、随机执行策略的系统奖励分别为 225.08、195.54、90.46。

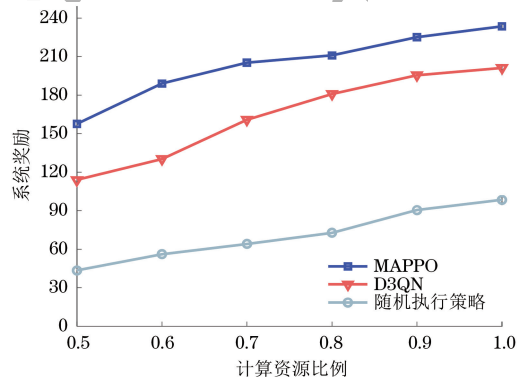


图 5 不同计算资源下的系统奖励

Fig.5 System rewards under different computing resources

不同算法在不同任务计算资源需求下的平均时延情况如图 6 所示。从图 6 中可以看出,在不同算法下,随着任务计算资源需求的增加,任务处理的平均时延逐渐增加,这是由于 3 个计算域的网络资源容量一定,当任务计算资源需求增加时,计算任务处理的时间变长。本文所采用的基于 MAPPO 的计算任务调度和资源分配算法的平均时延最低,当任务计算资源需求为 1.15 GHz 时,基于 MAPPO 的算法、基于 D3QN 的算法、随机执行策略的平均时延分别为 0.62、0.84、1.99 s。

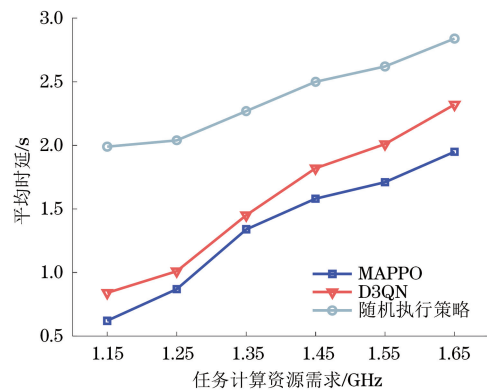


图 6 任务不同计算资源需求下的平均时延

Fig.6 Average delay of tasks under different computing resource requirements

4 结束语

本文深入探讨了 SAGIN 架构下的任务调度和资源分配问题,通过为地基、空基、天基网络分别部署相应的域控制器实现了 3 个计算域的分层控制,由此提出了基于 MAPPO 的任务调度和资源分配算法,3 个智能体与环境交互,协同决策,将计算任务调度到相应的计算域以达到算力资源的有效分配。实验结果表明,本文所提出的算法与其他算法相比在任务响应时间和能耗方面有较大的性能优势。

未来工作将研究 SAGIN 中的移动计算节点(尤其是无人机节点)的轨迹优化问题,以及任务调度时的确定性传输问题,这在任务数据收集和调度中发挥着重要作用。

参考文献

- [1] CHENG N, LÜ 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.
- [2] 张维庭, 孙呈蕙, 王洪超, 等. 算网资源智能适配与融合调度方法[J]. *电信科学*, 2023, 39(9): 12-20.
ZHANG W T, SUN C H, WANG H C, et al. Intelligent adaptation and integrated scheduling method for computing and networking resources[J]. *Telecommunications Science*, 2023, 39(9): 12-20. (in Chinese)
- [3] ZHANG P Y, WANG C, KUMAR N, et al. Space-air-ground integrated multi-domain network resource orchestration based on virtual network architecture: a DRL method[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(3): 2798-2808.
- [4] WANG C, LIU L, JIANG C X, et al. Incorporating distributed DRL into storage resource optimization of space-air-ground integrated wireless communication network[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2022, 16(3): 434-446.
- [5] ZHANG N, ZHANG S, YANG P, et al. Software defined space-air-ground integrated vehicular networks: challenges and solutions[J]. *IEEE Communications Magazine*, 2017, 55(7): 101-109.
- [6] HE J C, CHENG N, YIN Z S, et al. Service-oriented network resource orchestration in space-air-ground integrated network[J]. *IEEE Transactions on Vehicular Technology*, 2024, 73(1): 1162-1174.
- [7] GUO H Z, LI J Y, LIU J J, et al. A survey on space-air-ground-sea integrated network security in 6G [J]. *IEEE Communications Surveys & Tutorials*, 2022, 24(1): 53-87.
- [8] ZHANG P Y, CHEN N, SHEN S G, et al. AI-enabled space-air-ground integrated networks: management and optimization[J]. *IEEE Network*, 2024, 38(2): 186-192.
- [9] CAO B, ZHANG J T, LIU X, et al. Edge-cloud resource scheduling in space-air-ground-integrated networks for Internet of vehicles[J]. *IEEE Internet of Things Journal*, 2022, 9(8): 5765-5772.
- [10] 曾锋, 张政, 陈志刚. 基于深度强化学习的计算卸载与资源分配策略[J]. *通信学报*, 2023, 44(7): 124-135.
ZENG F, ZHANG Z, CHEN Z G. Computation offloading and resource allocation strategy based on deep reinforcement learning[J]. *Journal on Communications*, 2023, 44(7): 124-135. (in Chinese)
- [11] CHEN M Z, CHALLITA U, SAAD W, et al. Artificial neural networks-based machine learning for wireless networks: a tutorial[J]. *IEEE Communications Surveys & Tutorials*, 2019, 21(4): 3039-3071.
- [12] SUNDAR S, CHAMPATI J P, LIANG B. Multi-user task offloading to heterogeneous processors with communication delay and budget constraints[J]. *IEEE Transactions on Cloud Computing*, 2022, 10(3): 1958-1974.
- [13] ZHANG W T, YANG D, ZHANG C, et al. (Com)²Net: a novel communication and computation integrated network architecture[J]. *IEEE Network*, 2024, 38(2): 35-44.
- [14] TANG F X, HOFNER H, KATO N, et al. A deep reinforcement learning-based dynamic traffic offloading in Space-Air-Ground Integrated Networks (SAGIN)[J]. *IEEE Journal on Selected Areas in Communications*, 2022, 40(1): 276-289.
- [15] 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.
- [16] SHANG B D, YI Y, LIU L J. Computing over space-air-ground integrated networks: challenges and opportunities [J]. *IEEE Network*, 2021, 35(4): 302-309.
- [17] 张婷婷, 武楠, 姚海鹏. 天地融合网络智能组网体系架构研究[J]. *天地一体化信息网络*, 2022, 3(3): 47-55.
ZHANG T T, WU N, YAO H P. Research on intelligent networking architecture for the integrated space-terrestrial networks [J]. *Space-Integrated-Ground Information Networks*, 2022, 3(3): 47-55. (in Chinese)
- [18] 李斌, 刘文帅, 费泽松. 面向空地天地异构网络的边缘计算部分任务卸载策略[J]. *电子与信息学报*, 2022, 44(9): 3091-3098.
LI B, LIU W S, FEI Z S. Partial computation offloading for mobile edge computing in space-air-ground integrated network [J]. *Journal of Electronics & Information Technology*, 2022, 44(9): 3091-3098. (in Chinese)
- [19] PENG H X, SHEN X S. DDPG-based resource management for MEC/UAV-assisted vehicular networks[C]//*Proceedings of the IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. Washington D. C., USA: IEEE Press, 2020: 1-6.
- [20] CHENG M, ZHU C L, LIN M, et al. An O-MAPPO scheme for joint computation offloading and resources allocation in UAV assisted MEC systems [J]. *Computer Communications*, 2023, 208: 190-199.
- [21] XIAO Y, SONG Y Q, LIU J. Towards energy efficient resource allocation: when green mobile edge computing meets multi-agent deep reinforcement learning[C]//*Proceedings of the IEEE International Conference on Communications*. Washington D. C., USA: IEEE Press, 2022: 4056-4061.
- [22] JIANG Y Y, MAO Y X, WU G X, et al. A collaborative optimization strategy for computing offloading and resource allocation based on multi-agent deep reinforcement learning [J]. *Computers and Electrical Engineering*, 2022, 103: 108278.
- [23] YU C, VELU A, VINITSKY E, et al. The surprising

- effectiveness of PPO in cooperative, multi-agent games[EB/OL]. [2024-02-11]. <https://arxiv.org/abs/2103.01955v4>.
- [24] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB/OL]. [2024-02-11]. <https://arxiv.org/abs/1707.06347>.
- [25] SCHULMAN J, MORITZ P, LEVINE S, et al. High-dimensional continuous control using generalized advantage estimation[EB/OL]. [2024-02-11]. <https://arxiv.org/abs/1506.02438v6>.
- [26] KANG H Y, CHANG X L, MIŠIĆ J, et al. Cooperative UAV resource allocation and task offloading in hierarchical aerial computing systems: a MAPPO-based approach[J]. IEEE Internet of Things Journal, 2023, 10(12): 10497-10509.

编辑 陆燕菲

计算机工程
www.ecice06.com