计算机科学
COMPUTER SCIENCE

低轨卫星网络中基于深度强化学习的航空器任务卸载策略

李芳，袁宝淳，沈航，王天荆，白光伟

引用本文

李芳，袁宝淳，沈航，王天荆，白光伟. 低轨卫星网络中基于深度强化学习的航空器任务卸载策略[J]. 计算机科学，2026，53(2)：406-415.

LI Fang, YUAN Baochun, SHEN Hang, WANG Tianjing, BAI Guangwei. Deep Reinforcement Learning-based Aircraft Task Offloading in Low Earth Orbit Satellite Networks [J]. Computer Science, 2026, 53(2): 406-415.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250200092

低轨卫星网络中基于深度强化学习的航空器任务卸载策略

李 芳 袁宝淳 沈 航 王天荆 白光伟
南京工业大学计算机与信息工程学院(人工智能学院) 南京 211816
(202261220027@njtech.edu.cn)

摘 要 低地球轨道(LEO)卫星通信具有传输距离远、覆盖范围广、不受地形地貌限制等优点,已成为民航运输业和通用航空业的重要通信手段。然而,低轨卫星网络是一个高度异构和动态的环境,卫星节点的移动性、通信链路的复杂性、航空器时空分布不均和多种业务并存等特点使得任务卸载和资源分配面临许多挑战性问题。为此,提出了一种基于双深度强化学习(Double Deep Reinforcement Learning,DDRL)的航空器任务卸载方法,目的是最大化系统整体效用。首先,系统效用最大化问题被建模为一个任务卸载和资源分配的联合优化问题,同时考虑LEO卫星的计算能力和覆盖时间。接下来,将问题转换为马尔可夫决策过程,利用双重深度Q网络(Dual Deep Q Network,DDQN)算法学习最优的任务卸载决策,并在此基础上使用时间差分三重策略梯度(Time Difference Triple Policy Gradient,TD3)算法以获得最优资源分配策略。仿真实验表明,在不同的计算资源和通信资源下,所提出的方案在系统效用上优于其他基准方案,证明了所提框架的可用性。

关键词:低轨卫星网络;任务卸载;资源分配;深度强化学习
中图分类号 TP393

Deep Reinforcement Learning-based Aircraft Task Offloading in Low Earth Orbit Satellite Networks

LI Fang,YUAN Baochun,SHEN Hang,WANG Tianjing and BAI Guangwei
College of Computer and Information Engineering(College of Artificial Intelligence),Nanjing Tech University,Nanjing 211816,China

Abstract LEO satellite communication has the advantages of long transmission distance,wide coverage,and is not restricted by terrain. It has become an important communication method for the civil aviation transportation and general aviation industries. However,the low-orbit satellite network is a highly heterogeneous and dynamic environment. The mobility of satellite nodes,the complexity of communication links,uneven spatial and temporal distribution of aircraft,and the coexistence of multiple services make task offloading and resource allocation face many challenges. To this end,this paper proposes an aircraft task offloading method based on DDRL,with the purpose of maximizing the overall effectiveness of the system. Firstly,the system utility maximization problem is modeled as a joint optimization problem of task offloading and resource allocation,taking into account the computing power and coverage time of LEO satellites. Next,the problem is transformed into a Markov decision process,using DDQN algorithm to learn the optimal task offloading decision,and based on this,TD3 is used to obtain the optimal resource allocation strategy. Simulation experiments show that under different computing resources and communication resources,the proposed scheme is better than other benchmark schemes in terms of system utility,proving the usability of the proposed framework.

Keywords LEO satellite network,Task offloading,Resource allocation,Deep reinforcement learning

1 引言

低地球轨道(Low Earth Orbit,LEO)卫星通信具有传输距离远、时延和成本低、不受地形地貌限制等优点,可以有效解决以民航客机为代表的民用航空器在航行过程中的“航程信息孤岛”问题。随着业务数据量越来越多,传统卫星网络将数据回传至地面进行处理的中心式处理的计算时延越来越长,与业务的低时延要求形成矛盾。边缘计算的出现为用户利用网络边缘节点的计算能力进行数据处理提供了新的模式[1]。在边缘计算模式下,将每颗卫星视为边缘节点,实现在

到稿日期:2025-02-24 返修日期:2025-05-19
基金项目:国家自然科学基金(61502230,61501224);江苏省自然科学基金(BK20201357);江苏省“六大人才高峰”高层次人才项目(RJFW-020);江苏省研究生科研与实践创新计划(SJCX24_0566)
This work was supported by the National Natural Science Foundation of China(61502230,61501224),Natural Science Foundation of Jiangsu Province,China(BK20201357),Six Talent Peaks Project in Jiangsu Province(RJFW-020) and Postgraduate Research & Practice Innovation Program of Jiangsu Province(SJCX24_0566).
通信作者:沈航(hshen@njtech.edu.cn)

轨边缘计算，通过在数据源处处理数据，可以降低卫星数据传输带来的延迟[2]。

在上述背景下，文献[3-6]从架构和应用上对天基边缘计算进行了研究，提出了卫星边缘计算架构，将卫星边缘计算系统分为用户节点、卫星边缘节点和地面数据中心 3 部分，可以减少业务时延以及节省带宽。Wang 等[7]进一步介绍了将传统卫星改造成空间边缘计算节点相应的卫星硬件结构和软件架构，通过卫星资源的虚拟化，实现灵活的计算服务。Xie 等[8]对天地一体化边缘计算网络中的协同计算卸载、多节点任务调度和移动性管理等进行了全面详尽的分析，并指出卫星边缘计算面临的技术挑战。Li 等[9]考虑卫星是具有计算和存储资源的边缘计算平台，通过解决混合整数线性规划问题来解决系统服务请求调度决策和服务放置问题。Cao 等[10]提出了一种基于软件定义网络和网络功能虚拟化的卫星边缘云架构，提出同时考虑能源消耗、服务延迟和资源利用的计算资源调度方法。由于卫星平台的通信、功耗、计算资源均受限，文献[11-14]研究联合计算通信资源分配和卸载决策的卸载算法，分别基于深度强化学习、博弈论和启发式搜索算法实现卸载决策，以缩短系统服务时延和降低功耗。

然而，低轨卫星网络仍然面临挑战。尽管在 LEO 卫星上部署移动边缘计算(Mobile Edge Computing，MEC)服务器极大地增强了其计算能力，但受限于卫星的体积和能源供应，每颗卫星能够提供的计算资源仍然有限。通过合理安排卫星星座内的计算资源，可以把任务转移给那些空闲资源较多的卫星进行处理。通过高效利用有限的卫星资源，不仅能快速响应，满足用户对低延迟的要求，也能降低单个卫星承担的工作负荷，减少能耗。因此，如何通过对网络边缘分散的卫星资源进行合理分配和调度，实现航空器任务直接卸载到低轨卫星进行处理，降低任务传输过程中的时延和能耗，成为卫星边缘计算的挑战之一。

针对上述挑战，本文设计了一种基于双深度强化学习(DDRL)框架的任务卸载和资源分配联合优化方案，目的是最大化系统效用。本文的主要贡献如下：

1)针对 LEO 卫星网络中的任务卸载和资源分配问题，考虑到低轨卫星移动性、卫星自身计算能力差异、多用户任务等因素，以最大化系统效用为目标，提出了一种任务卸载和资源分配模型。

2)考虑到决策卸载和资源分配是一个离散变量和连续变量混合的复杂优化问题，利用双重深度 Q 网络(DDQN)算法对系统环境进行学习，得到该环境下的最优决策卸载，再采用时间差分三重策略梯度(TD3)算法对资源进行最优分配。

3)在卫星环境仿真平台模拟 LEO 卫星网络环境，以验证 DDRL 的性能。仿真实验表明，该方案与基准算法相比，能够有效地访问和协同计算并发任务，并且在不同的环境变量下具有更好的收敛性和优越性。

2　系统模型和问题建模

本章首先描述 LEO 卫星网络系统的网络模型；其次，建立基于航空用户与卫星关联的任务卸载和资源分配的通信模型和计算模型；最后，提出了在约束条件下实现系统能耗最小化的优化问题。表 1 列出了本文后续将用到的关键符号和相关描述。

表 1　重要符号

Table 1　Important symbols

符号	定义
$\mathcal{U}/U$	用户集合/用户数量
S/S	成员卫星集合/成员卫星数量
Y	系统时隙数量
τ	系统时间的时隙长度
μ	任务到达时间间隔的平均值
$l_{u,t}$	用户 u 在时隙 t 的任务数据大小
$c_{u,t}$	用户 u 在时隙 t 的任务计算负载
$T_{u,t}^{(\max)}$	用户 u 在时隙 t 的任务最大可容忍时延
$O_{u,s}^{(t)}$	用户 u 在时隙 t 的任务卸载决策
$F_{u,s}^{(t)}$	卫星 s 为用户 u 在时隙 t 的任务的计算资源分配策略
$f_s^{(\max)}$	成员卫星 s 拥有的最大计算资源
κ	有效电容系数
p_u	用户 u 的终端发射功率
p	簇首卫星的发射功率
$h_{u,t}$	簇首卫星与用户 u 在时隙 t 的信道条件
B	簇首卫星与关联用户之间的总带宽
$d_{u,t}$	簇首卫星与用户 u 在时隙 t 的距离
ω	ISL 通信能力
$\boldsymbol{\varphi}_{s,t}$	成员卫星 s 在时隙 t 与簇首卫星之间的距离

2.1　网络模型

如图 1 所示，在 LEO 卫星网络系统架构中，LEO 卫星星座被划分为若干个管理域，每个管理域是一个子网络系统。管理域由 1 个簇首卫星和S个搭载 MEC 服务器的簇内成员卫星组成，簇内成员卫星用集合S表示，其中成员卫星 $s\in S$。簇首卫星与成员卫星之间以及部分成员卫星之间存在星间链路(Inter Satellite Link，ISL)。固定卫星管理节点面临能量损耗、空间位置变化等情况，且后期存在补网、功能更新等。当前簇首卫星控制管理能力存在波动，需周期性地进行簇首选举，以最大程度地发挥系统中簇首卫星的管控能力。因此，本文采用文献[15]中“静态指数＋动态指数”的多因素加权簇首选举办法定期选举簇首卫星。航空器用户用集合$\mathcal{U}$表示，以民航飞机为例，其中航空器用户 $u\in\mathcal{U}$。航空器用户的个数用$\mathcal{U}$表示。假设此时航空器位于偏远区域，它只能通过接入 LEO 卫星移动边缘计算网络获取服务[16]。当航空器进入一个 LEO 卫星星座管理域时，会自动接入该管理域的簇首卫星，簇首卫星根据任务的类型将其卸载到成员卫星进行处理。需要说明的是，如果航空器被多个管理域所覆盖，则默认选择信道质量高的管理域簇首卫星接入。

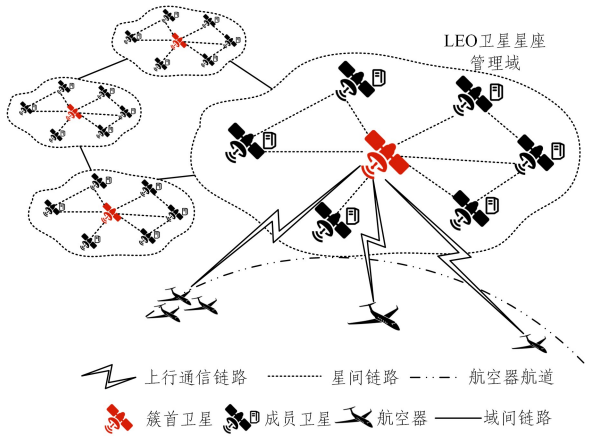


图 1　LEO 卫星网络系统

Fig. 1　LEO satellite network system

LEO 卫星网络系统以时隙模式运行，用集合 $\mathcal{T}$ 表示，时间隙 $t\in\mathcal{T}$。时间被划分为长度相等的 Y 个时隙，每个时隙长度为 τ。假设 LEO 星座的路由表和轨迹信息为簇首卫星已知。卫星之间的链路状态通常取决于卫星与地球之间的相对位置[17]，且当卫星位于地球的非两极区域时，ISL 的链路状态相对稳定。据此，为了简化模型，本文认为在一定时间内 LEO 卫星星座具有稳定的拓扑结构与连通性[18]。在每个时隙开始时，航空器会随机且并行地产生任务，这些任务独立且不可分割。假设任务到达的时间间隔遵循平均值为 μ 的指数分布。在时隙 t 中，用户 u 在时隙 t 生成的任务用三元组 $K_{u,t}=\{l_{u,t},c_{u,t},T_{u,t}^{(\max)}\}$ 表示，其中，$l_{u,t}$ 和 $c_{u,t}$ 表示任务的数据大小和工作负载，$T_{u,t}^{(\max)}$ 是任务的最大可容忍时延。此外，系统运行过程中，用户上传的总任务数量用 K 表示。任务的输出数据与输入数据相比通常很小[19]，因此可以忽略返回计算结果的延迟。此外，假设簇首卫星将任务卸载给成员卫星时，用户身份、通信协议等信息通常会作为任务分配消息的一部分被一起发送，在完成数个时隙的处理后，成员卫星可依据这些信息进行配置，建立起与用户的通信链路，然后返回处理结果。

如图 2 所示，LEO 卫星网络系统中任务卸载和资源分配的模型包括航空器接入并将任务传输至簇首卫星进行排队、簇首卫星从等待队列中卸载任务至成员卫星，以及在成员卫星上进行任务处理 3 个步骤。

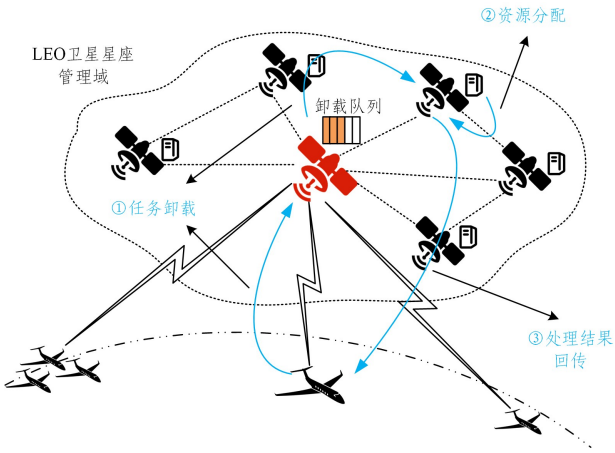


图 2 任务卸载与资源分配模型

Fig. 2 Task offloading and resource allocation model

首先，当簇首卫星接收到大量用户任务时，需要考虑任务分发卸载时的等待队列。簇首卫星将需要卸载的任务放入卸载队列 $N(t)$ 中。其次，设簇首卫星将任务卸载到成员卫星的决策用矩阵符号 $\boldsymbol{O}_{u,s}^{(t)}$ 表示。矩阵 $\boldsymbol{O}_{u,s}^{(t)}$ 表示是否将第 i 个用户的任务卸载到第 j 个卫星上处理，$o_{i,j}^{(t)}=1$ 表示将第 i 个用户的任务卸载到第 j 个卫星上处理，$o_{i,j}^{(t)}=0$ 表示不将第 i 个用户的任务卸载到第 j 个卫星上处理。最后，当任务被簇首卫星卸载到成员卫星 s 上后，s 需要为卸载任务 $\mathcal{K}_{u,t}$ 分配计算资源。任务 $\mathcal{K}_{u,t}$ 的计算资源分配策略用 $F_{u,s}^{(t)}$ 表示。它为任务 $\mathcal{K}_{u,t}$ 提供的计算资源大小为 $f_{u,s}^{(t)}$，且 $0\leqslant f_{u,s}^{(t)}\leqslant f_s^{(\max)}$，其中 $f_s^{(\max)}$ 表示一颗成员卫星 s 所拥有的最大计算资源。此外，还采用了基于预订的资源分配机制[20]。一个任务被卸载到卫星上后，为该任务分配的正交计算资源将被保留和占用几个时间段，直到任务计算完成。这种方式可以确保任务在到达卫星时立即得到处理，且没有额外的延迟抖动[21]。因此，由于不同卫星上任务处理进程的不同，卫星对当前任务的可用计算资源也有所不同。卫星在时隙 t 分配的计算资源不能超过当前可用资源，如式(1)所示：

$$\sum_u f_{u,s}^{(t)}\leqslant f_s^{(\max)}-\boldsymbol{\phi}_{s,t} \tag{1}$$

其中，$\boldsymbol{\phi}_{s,t}$ 表示卫星 s 在时隙 t 已经被占用的资源。

2.2 通信模型

1)卫星-航空器通信模型

用户通过与簇首卫星之间的链路上传任务，传输链路使用 Ka 频段。这里根据文献[22]，在一个时隙内考虑准静态衰落信道模型。具体来说，簇首卫星在时隙 t 从用户 u 接收到的信号 $y_{u,t}$ 表示为：

$$y_{u,t}=\sqrt{p_u}h_{u,t}s_u+N_0 \tag{2}$$

其中，p_u 和 s_u 分别代表用户 u 的发射功率和数据信号；$N_0\sim(0,\sigma^2)$ 是信道的加性白高斯噪声(Additive White Gaussian Noise，AWGN)，σ^2 是噪声方差。$h_{u,t}$ 表示时隙 t 中用户 u 和簇首卫星之间的 AWGN 信道，其表示为：

$$h_{u,t}=\delta_u\eta_u\,(d_{u,t})^{-\beta} \tag{3}$$

其中，$\delta_u\sim(0,1)$ 是具有瑞利衰落的复高斯变量；η_u 是遵循对数正态分布的阴影衰落；$d_{u,t}$ 是时隙 t 中用户 u 与簇首卫星之间的实际距离，由用户 u 与簇首卫星之间的几何关系求得，如图 3 所示。由于低轨卫星的高度与一般航空器的高度之间存在两个及以上数量级的差距，因此本文对航空器的离地高度不予考虑，同理也不考虑航空器的速度。假设用户只能在一定的仰角 α 下开始传输请求，α 是相对于地心和用户位置的连线对称的最小角度。设 θ 表示卫星覆盖的弧度对应的中心角，R 为地球半径，H 为卫星轨道高度。在忽略其他因素的前提下，假设 α 已知，则 θ 和 α 之间的关系可以表示为：

$$\theta=\alpha-2\arcsin\left(\frac{R\cos(\alpha/2)}{R+H}\right) \tag{4}$$

由此，簇首卫星与用户 u 在时隙 t 时的实际距离为：

$$d_{u,t}=(R+H)\frac{\sin\theta}{\sin(\alpha/2)} \tag{5}$$

此外，β 是路径损耗指数，这意味着卫星与用户之间的通信条件与两者之间的距离成负相关。

为了简化问题，假设卫星的频谱对所有用户进行正交分配，且簇首卫星的总通信带宽被当前所关联的用户平均分配。因此，根据香农公式，用户 u 到簇首卫星的数据传输率可以表示为：

$$R_{u,t}=\frac{B}{U}\log_2\left(1+\frac{p_u|h_{u,t}|}{\sigma^2}\right) \tag{6}$$

其中，B 代表 Ka 频段上的总带宽。显然，更差的信道 $h_{u,t}$ 及更多的关联用户数量 U 会使得数据传输更慢。

考虑到延迟问题，由于数据包被推送至链路时机器内部会产生通信延迟，且电子信号进行发射时机器外部会产生传播延迟。设 c 表示光速，则用户 u 在时隙 t 与簇首卫星之间的时间延迟为：

$$T_{u,t}^{(1)}=\frac{l_{u,t}}{R_{u,t}}+\frac{d_{u,t}}{c} \tag{7}$$

此外，任务 $K_{u,t}$ 从用户 u 传输到簇首卫星的传输能耗为：

$$E_{u,t}^{(1)}=p_u\frac{l_{u,t}}{R_{u,t}} \tag{8}$$

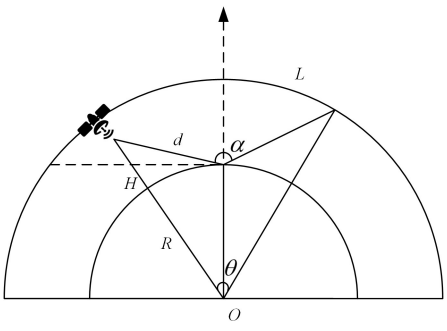


图 3　卫星与用户之间的几何关系

Fig. 3　Geometric relationship between satellite and user

2）卫星间通信模型

用 $N(t)$ 存储在时隙 t 中被上传到簇首卫星上还未卸载的任务。下一个时隙中的卸载队列 $N(t+1)$ 通过减去已卸载的任务数据量 $B(t)$，加上新上传到簇首卫星的任务数据量 $A(t,\mathcal{K}_{u,t})\tau$，得到：

$$N(t+1)=\max[N(t)-B(t),0]+A(t,\mathcal{K}_{u,t})\tau \tag{9}$$

其中，$A(t,\mathcal{K}_{u,t})$ 表示与时间 t 和任务 $\mathcal{K}_{u,t}$ 相关的服从泊松分布的任务到达率，τ 是一个时隙的长度。假设簇首卫星在时隙 t 处理卸载队列时 CPU 频率为 f'，则一个时隙内，簇首卫星处理卸载队列的任务数据量为：

$$B(t)=\frac{f'}{J}\tau \tag{10}$$

其中，J 表示处理每比特任务数据所需的 CPU 周期数。因此，卸载队列 $N(t)$ 的队列延迟可以表示为[23]：

$$T_t^{(2)}=\frac{X_{N(t)}}{\tilde{r}_{N(t)}} \tag{11}$$

其中，$X_{N(t)}$ 表示队列长度，$\tilde{r}_{N(t)}$ 表示队列 $N(t)$ 的平均任务到达率，如式(12)所示：

$$\tilde{r}_{N(t)}=\frac{1}{t}\sum_{m=0}^{t-1}A(t,\mathcal{K}_{u,t}) \tag{12}$$

另外，任务排队过程中不考虑能量消耗。

为提高实时性，假设任务只能从簇首卫星到成员卫星 s 完成一次卸载，不存在成员卫星间的多跳卸载过程。在簇首卫星做出卸载决策后，任务 $\mathcal{K}_{u,t}$ 会从簇首卫星的卸载队列中卸载到成员卫星进行处理。因此，用户 u 的任务在时隙 t 从簇首卫星到卸载卫星的时间延迟表示为：

$$T_{u,t}^{(3)}=\frac{l_{u,t}}{\omega}+\frac{\varphi_{s,t}}{c} \tag{13}$$

其中，$\varphi_{s,t}$ 表示在时隙 t 成员卫星 s 与簇首卫星之间的动态距离。同时，设 p 为簇首卫星的发射功率，则任务 $\mathcal{K}_{u,t}$ 的传输能耗可表示为：

$$E_{u,t}^{(2)}=p\frac{l_{u,t}}{\omega} \tag{14}$$

2.3　计算模型

在时隙 t 中，卸载卫星 s 分配给用户 u 的计算资源为 $f_{u,s}^{(t)}$，由此可知，处理任务 $\mathcal{K}_{u,t}$ 的计算时间为：

$$T_{u,t}^{(4)}=\frac{l_{u,t}c_{u,t}}{f_{u,s}^{(t)}} \tag{15}$$

本阶段主要考虑任务的计算时延，当前任务计算前的排队时延忽略不计。

本文使用了一个被广泛采用的基于一个计算周期的能耗模型，卫星上一个任务的处理能耗被计算为 $e=\kappa f^2$[24]，其中 κ 是能耗系数，取决于芯片架构的有效开关电容，而 f 是 CPU 频率。因此，在卸载卫星 s 上执行任务 $\mathcal{K}_{u,t}$ 的计算能耗表示为：

$$E_{u,t}^{(3)}=\kappa(f_{u,s}^{(t)})^2 l_{u,t}c_{u,t} \tag{16}$$

2.4　问题建模

基于上述讨论，任务 $\mathcal{K}_{u,t}$ 从接入到卸载的总时延 $T_{u,t}$ 由式(7)、式(11)、式(13)和式(15)组成，表示为：

$$T_{u,t}=\sum_{i\in\{1,2,3,4\}}T_{u,t}^{(i)} \tag{17}$$

且 $T_{u,t}\leqslant T_{u,t}^{(\max)}$，即每个任务的端到端延迟不能超过最大可容忍时延。将该条件作为任务完成与否的标志，用二元变量 $a_{u,t}$ 表示，定义为：

$$a_{u,t}=\begin{cases}1, & \text{if } T_{u,t}\leqslant T_{u,t}^{(\max)}\\ 0, & \text{otherwise}\end{cases} \tag{18}$$

如果能满足延时约束，则认为该任务被成功处理，此时 $a_{u,t}=1$；否则认为任务失败，此时 $a_{u,t}=0$。

这个过程中产生的系统能耗由式(8)、式(14)和式(16)组成，表示为：

$$E_{u,t}=\sum_{i\in\{1,2,3\}}E_{u,t}^{(i)} \tag{19}$$

因此，任务 $\mathcal{K}_{u,t}$ 的端到端时延及系统能耗由任务卸载和资源分配的联合策略决定。每个时隙开始时，簇首卫星决定将任务卸载到哪一颗成员卫星以及为该任务分配多少计算资源。

系统成本包含系统运行过程中所有任务的时延和能耗的求和，任务卸载和资源分配的联合优化可以描述为系统成本的最小化问题，表示为：

$$\begin{aligned}\min_{O_{u,s}^{(t)},F_{u,s}^{(t)}}\ &(\xi\sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}a_{u,t}\log(1+E_{u,t})+\\ &(1-\xi)\sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}a_{u,t}\log(1+T_{u,t}))\end{aligned} \tag{20}$$

其中，$\xi\in[0,1]$ 用于权衡能耗成本和时延成本在系统成本中的比重。

将系统成本的最小化转换为系统效用的最大化问题，建模为：

$$\begin{aligned}\text{P1}:\ \max_{O_{u,s}^{(t)},F_{u,s}^{(t)}}\ &(\xi\sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}a_{u,t}\log(1+\lambda-E_{u,t})+\\ &(1-\xi)\sum_{t\in\mathcal{T}}\sum_{u\in\mathcal{U}}a_{u,t}\log(1+\lambda-T_{u,t}))\end{aligned} \tag{21}$$

$$\text{s.t. } o_{i,j}^{(t)}\in\{0,1\},\forall i\in\mathcal{U},\forall j\in\mathcal{S},\forall t\in\mathcal{T} \tag{21a}$$

$$\sum_{j=1}^{S}o_{i,j}^{(t)}=1,\forall i\in\mathcal{U},\forall t\in\mathcal{T} \tag{21b}$$

$$0<f_{u,s}^{(t)}\leqslant f_s^{(\max)},\forall u\in\mathcal{U},\forall s\in\mathcal{S},\forall t\in\mathcal{T} \tag{21c}$$

$$\sum_u f_{u,s}^{(t)}\leqslant f_s^{(\max)}-\phi_{s,t},\forall s\in\mathcal{S},\forall t\in\mathcal{T} \tag{21d}$$

其中，λ 为开销因子，用来解决任务未完成导致任务时延和能耗无法计入的问题，其为定值且充分大。约束条件(21a)和(21b)表明每个用户至多可以将一个任务卸载给一颗 LEO

卫星;(21c)表示卫星分配给任务的资源不能超过卫星所拥有的最大资源;(21d)表示每个卫星分配给当前卸载任务的计算资源之和不得超过该卫星当前的可用资源。

3 算法设计

根据上文的讨论,$o_{u,s}^{(t)}$是离散变量,$f_{u,s}^{(t)}$是连续变量,本文需解决的是一个离散变量和连续变量混合的复杂优化问题。因此,本章提出一种 DDRL 框架,用来解决联合决策问题。针对离散变量环境,利用 DDQN 对系统环境进行学习,得到该环境下的最优决策卸载,再针对连续变量环境,采用 TD3 对资源进行分配,以得到最优的资源分配策略。

3.1 问题转化

考虑到决策卸载与资源分配问题具有关联性,即资源分配与任务卸载耦合,本文将这两个问题进行联合处理。

由于相邻时隙决策的相关性,上述动态场景中的任务卸载和资源分配过程可以转换为一个马尔可夫决策过程(Markov Decision Process,MDP)问题。MDP 由 5 个元素构成:状态空间 S、行动空间 A、状态转移概率 P、奖励函数 R 以及折扣因子 γ。在时隙 t,簇首卫星可以获取到成员卫星的负载情况、剩余资源等状态信息及上传的用户任务信息。然后簇首卫星输出任务卸载和资源分配决策,并在做出卸载动作的同时将资源分配决策返回给成员卫星,成员卫星分配资源进行任务处理。最后,成员卫星将结果返回给用户。簇首卫星通过与系统环境的不断交互来学习和实现策略,并获得奖励以改进策略。对 MDP 中关键元素的描述如下:

状态空间:在时隙 t,簇首卫星从整体的角度掌握所有航空器用户和簇成员卫星的状态信息。系统状态表示为 $S_t=\{\boldsymbol{L}_{u,t},\boldsymbol{D}_{u,t},K_{u,t},\boldsymbol{\varphi}_{s,t},\boldsymbol{\phi}_{s,t},N(t)\}$。其中,$\boldsymbol{L}_{u,t}$表示用户 u 在时隙 t 的地理位置,位置信息被描述为一个二维向量,包含大地坐标系下的经度和纬度,高程设为 0;$\boldsymbol{D}_{u,t}$是一个 U 维向量,每个元素是用户 u 在时隙 t 与簇首卫星之间的距离;$K_{u,t}$表示用户 u 在时隙 t 上传任务的属性信息;$\boldsymbol{\varphi}_{s,t}$是一个 S 维向量,是指卫星 s 在时隙 t 与簇首卫星之间的距离;$\boldsymbol{\phi}_{s,t}$也是一个 S 维向量,表示每颗成员卫星当前已被占用的资源数量;$N(t)$表示当前卸载队列的排队情况。

动作空间:动作指簇首卫星在时隙 t 做出卸载卫星的选择和资源分配方案,将其定义为 $A_{u,t}=\{a_{u,t}^{(O)},a_{u,t}^{(F)}\}$。其中,$a_{u,t}^{(O)}$和 $a_{u,t}^{(F)}$分别表示任务卸载动作和资源分配动作。在动态任务卸载和资源分配过程中,簇首卫星需要观察系统环境 S_t 确定卸载卫星,并在做出卸载决策的基础上综合系统环境观察信息 S_t,确定卸载卫星为任务 $\mathcal{K}_{u,t}$ 分配的计算资源。

奖励:在系统状态 S_t 下对用户任务 $\mathcal{K}_{u,t}$ 采取动作 $A_{u,t}$ 后,系统环境会进入一个新的状态 S_{t+1} 并返回相应的奖励。将系统在状态 S_t 下的总成本用符号 $\psi(S_t)$ 表示,当系统经动作 $A_{u,t}$ 进入状态 S_{t+1} 后,状态 S_t 和 S_{t+1} 存在成本差 $\psi(S_t)-\psi(S_{t+1})$。因此,奖励函数的设计基于系统状态转换过程的成本差,具体公式为:

$$r(S_t,A_t,S_{t+1})=\psi(S_t)-\psi(S_{t+1}) \tag{22}$$

当 $r(S_t,A_t,S_{t+1})>0$,即系统成本随时隙减少时,该动作会得到一个正向的奖励值;反之,奖励值为负数,表示该动作受到惩罚。

3.2 模型训练

所提 DDRL 框架包含 DDQN 和 TD3 两个模块。其中,DDQN 用于卸载决策问题,TD3 用于计算资源分配问题。用户任务与环境的详细交互过程如图 4 所示,共分为 5 个步骤。

步骤 1 DDQN 以环境状态作为输入,迭代地为每个用户提供服务。

步骤 2 神经网络处理信息并为每个用户输出适当的卸载决策。

步骤 3 环境状态和用户的卸载决策被 TD3 算法作为输入进行处理。

步骤 4 TD3 考虑以上信息,然后给出合适的计算资源分配。

步骤 5 DDRL 输出最后的联合决策。

具体来说,每个交互首先通过 DDQN 执行任务卸载动作 $a_{u,t}^{(O)}$,然后根据 TD3 执行资源分配动作 $a_{u,t}^{(F)}$。另外,任务 $\mathcal{K}_{u,t}$ 从状态 S_t 经动作 $A_{u,t}$ 进入状态 S_{t+1} 后的奖励值 $r(S_t,A_t,S_{t+1})$用符号 $r_{u,t}$ 表示。当 DDQN 与环境交互时,它遵循 ε-greedy 策略来执行卸载决策的动作。ε 是一个概率决策因子,介于 0 到 1 之间。在 DDQN 学习过程中,ε 表示随机选择一个动作的概率,$1-\varepsilon$ 表示执行由 DDQN 主网络给出的动作 $a_{u,t}^{(O)}$ 的概率。

刚开始探索环境时,会给 ε 设置一个较大值,这样可以带来更多的随机性,有利于发现新的状态和动作。随着学习的积累,ε 值会逐渐减小,智能体将更多地利用已学到的知识,选择看上去回报最大的动作。另一方面,由于 TD3 适用于连续动作 $a_{u,t}^{(F)}$,因此其探索方法会添加随机噪声。该噪声服从具有方差 σ 的正态分布,σ 值在学习过程中逐渐减小。

在完成交互后,算法 1 会生成一个用于 DDQN 的观察值和一个用于 TD3 的观察值。这些观察值会分别存储在两个独立的回放缓冲区 B_1 和 B_2 中。

算法 1 任务与环境交互的样本生成

输入:成员卫星、用户、任务及队列的信息,当前时间步 t

输出:每个任务与环境交互的样本数据

1. 初始化回放缓冲区 B_1 和 B_2;
2. for k=1,2,…,K do
3. 根据系统环境信息初始化状态 S_t;
4. 计算状态 S_t 下每个动作 $a_{u,t}^{(O)}$ 的概率向量 $\mathbf{Q}(S_t)$;
5. 删除 $\mathbf{Q}(S_t)$中不可行的动作,得到新的向量 $\mathbf{Q}'(S_t)$;
6. 基于 ε-greedy 策略从 $\mathbf{Q}'(S_t)$中选取卸载动作 $a_{u,t}^{(O)}$;
7. 基于 $A(S_t,a_{u,t}^{(O)})+N(0,\sigma)$选取资源分配动作 $a_{u,t}^{(F)}$;
8. 任务 $K_{u,t}$基于动作组$(a_{u,t}^{(O)},a_{u,t}^{(F)})$与环境交互,得到奖励值 $r_{u,t}$和下一个状态 S_{t+1};
9. 将四元组$(S_t,a_{u,t}^{(O)},S_{t+1},r_{u,t})$存储到回放缓冲区 B_1 中;
10. 将五元组$(S_t,a_{u,t}^{(O)},a_{u,t}^{(F)},S_{t+1},r_{u,t})$存储到回放缓冲区 B_2 中;
11. end for
12. return 样本数据$(S_t,a_{u,t}^{(O)},a_{u,t}^{(F)},S_{t+1},r_{u,t})$。

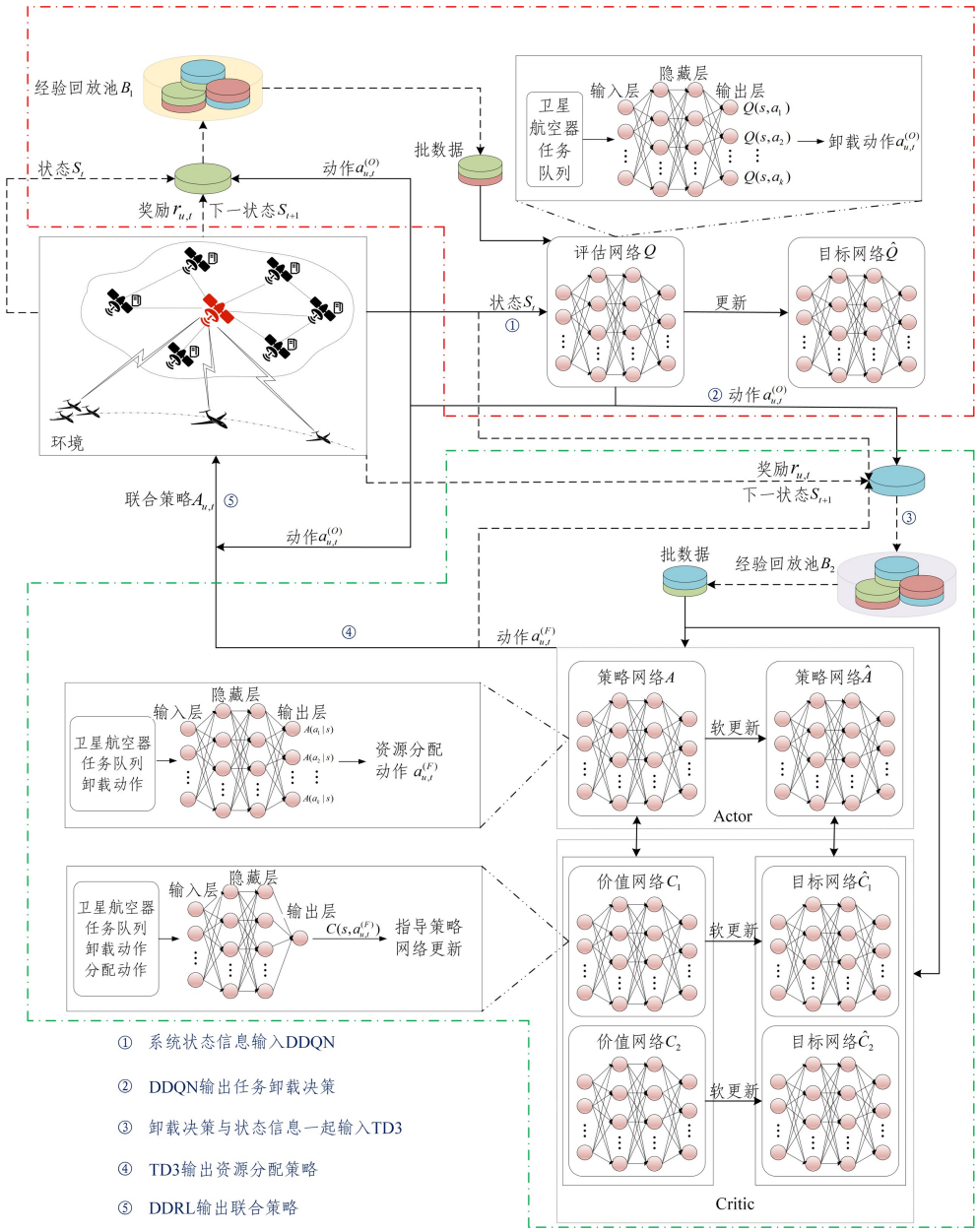


图 4　DDQN 与 TD3 协同训练框架

Fig. 4　Collaborative training framework of DDQN and TD3

3.3　DDRL 联合优化框架

基于 DDRL 的任务卸载与资源分配被描述为算法 2，其样本数据采集自算法 1。算法 2 的第一部分是 DDQN 算法的更新过程。这一部分设置了 Q 网络和目标网络 $\hat{Q}$。DDQN 从回放缓冲区 B_1 中采集四元组$(S_t, a_{u,t}^{(O)}, S_{t+1}, r_{u,t})$，用于 Q 和 $\hat{Q}$ 的学习。当回放缓冲区 B_1 满时，最新的观测结果将取代旧观测结果。在使用样本更新 Q 时，DDQN 从 $\hat{Q}$ 中计算值 $y_{u,t}$ 作为目标值，表示为：

$$y_{u,t}=r_{u,t}+\gamma\hat{Q}(S_{t+1},\arg\max_{a_{u,t}^{(O)}}Q(S_{t+1},a_{u,t}^{(O)})) \tag{23}$$

其中，γ 为衰减因子。更新 Q 的损失函数是平方误差函数，定义为：

$$Loss=\sum_{(S_t,a_{u,t}^{(O)},S_{t+1},r_{u,t})\in B_1}(y_{u,t}-Q(S_t,a_{u,t}^{(O)}))^2 \tag{24}$$

接着，使用式(24)计算梯度，并每隔 g_1 步更新一次 Q 网络。同时，通过设置 $\hat{Q}=Q$，每隔 G_1 步更新一次 $\hat{Q}$。

算法 2 的第二部分是 TD3 算法的更新过程。在对任务做出卸载决策后，该算法为任务分配最优的计算资源数量。TD3 是一种 Actor-Critic 方法，包含两个独立的价值网络和一个策略网络及其对应的目标网络。Actor 网络 A 预测资源分配策略 $a_{u,t}^{(F)}$，这是一个计算资源的连续值；Critc 网络 C_1 和 C_2 判断在学习过程中由 Actor 网络生成的策略的 q 值。在训练 TD3 之前，算法会从回放缓冲区 B_2 中采样部分值。在训练过程中，该算法会从目标网络 $\hat{A}$ 和 $\hat{C}$ 中计算目标值：

$$y_{u,t}=r_{u,t}+\gamma\min(\hat{C}_1(S_{t+1},a_{u,t}^{(F)}+\varepsilon),\hat{C}_2(S_{t+1},a_{u,t}^{(F)}+\varepsilon)) \tag{25}$$

其中，$\varepsilon\sim\mathrm{clip}(\mathcal{N}(0,\sigma,-c,c))$。然后，将目标值 $y_{u,t}$ 与 q 值进行比较，两者之间的差值为时间差值(TD)误差 δ。该函数是一个平方误差，定义为：

$$\delta_l = \sum_{(S_t, a_{u,t}^{(O)}, a_{u,t}^{(F)}, S_{t+1}, r_{u,t}) \in B_2} (y_{u,t} - C_l(S_t, a_{u,t}^{(F)}))^2, l \in \{1,2\} \tag{26}$$

式(26)的梯度将用于更新 Critic 网络 C_1, C_2。为了更新 Actor 网络 A,算法计算从 Critic 网络 C_1 预测的 $q_1 = C_1(s,a)$ 值的梯度。由于 Actor 网络 A 在更新 Critic 网络 C_1 时保持固定,因此算法使用梯度上升法(最小化 $-q_1$)来最大化 q_1 值。训练网络 A, C_1 和 C_2 每 g_2 步更新一次,同时,目标网络 $\hat{A}$, $\hat{C}_1$ 和 $\hat{C}_2$ 每 G_2 步更新一次,具体做法是将目标网络的参数同步为当前训练网络的参数,以保持两者一致。此外,为了保证策略网络更新的稳定,每更新 μ 次价值网络才更新一次策略网络。

算法 2 基于 DDRL 的任务卸载与资源分配算法

输入:成员卫星、用户、任务及队列的信息

输出:每个任务的最优卸载和资源分配策略

1. 初始化 DDQN 推理网络 Q 和目标网络 $\hat{Q}$, TD3 推理网络 A, C_1, C_2 和目标网络 $\hat{A}$, $\hat{C}_1$, $\hat{C}_2$;
2. 初始化全局计数器 x=0 和回放缓冲区 B_1 和 B_2;
3. for 回合 episode←1 do
4. for t=1,2,…,Y−1 do
5. 使用算法 1 通过任务与环境之间的交互获取观察结果(S_t, $a_{u,t}^{(O)}$, $a_{u,t}^{(F)}$, S_{t+1}, $r_{u,t}$);
6. 将观测结果(S_t, $a_{u,t}^{(O)}$, S_{t+1}, $r_{u,t}$)和(S_t, $a_{u,t}^{(O)}$, $a_{u,t}^{(F)}$, S_{t+1}, $r_{u,t}$)分别添加到 B_1 和 B_2;
7. x←x+1;
8. if x%g_1== 0 do
9. 从 B_1 中采样小批量观测值;
10. 使用样本计算目标值 $y_{u,t} \leftarrow r_{u,t} + \gamma \hat{Q}(S_{t+1}, \arg\max_{a_{u,t}^{(O)}} Q(S_{t+1}, a_{u,t}^{(O)}))$;
11. 使用目标值 $y_{u,t}$ 查找梯度并更新 Q;
12. end if
13. if x%g_2== 0 do
14. 从 B_2 中采样小批量观测值;
15. 使用样本计算目标值 $y_{u,t} \leftarrow r_{u,t} + \gamma \min(\hat{C}_1(S_{t+1}, a_{u,t}^{(F)}), \hat{C}_2(S_{t+1}, a_{u,t}^{(F)}))$;
16. 使用样本从价值网络 C_1, C_2 计算 q_1, q_2 值;
17. 使用 $y_{u,t}$ 和 q_1, q_2 计算时间差分目标;
18. 使用 TD 值得到梯度并更新 C_1, C_2;
19. 使用均值 $-q_1$ 得到梯度并更新 A;
20. end if
21. if x%G_1== 0 do
22. $\hat{Q}$←Q;
23. end if
24. if x%G_2== 0 do
25. $\hat{C}_1$←C_1, $\hat{C}_2$←C_2;
26. $\hat{A}$←A
27. 降低参数 σ 值;
28. end if
29. 降低参数 ε 值;
30. end for
31. end for
32. return 最优任务卸载和资源分配策略 Q 和 A。

3.4 时间复杂度分析

首先,总训练回合数决定了算法需要迭代的次数。每一次迭代都意味着一次完整的训练过程,包括从环境中获取反馈并更新策略。显然,更多的训练回合会直接增加算法的总体计算量。本文用$|episode|$表示总训练回合数 $episode$ 的大小,则时间复杂度与$|episode|$成线性关系。其次,环境终止时间 Y 表示每个训练回合中环境运行的步数上限。每一步都涉及状态转移、动作选择以及奖励计算等过程。每增加一步,算法的计算量都会相应增加,所以时间复杂度也与 Y 成线性关系。最后描述用户任务的数量 K。K 代表了在每个训练步中需要处理的任务数量。在每一步中,算法需要为每个用户任务进行计算和决策。因此,K 的增加会线性地增加每一步的计算量,最终影响整体时间复杂度。综上所述,DDRL 的时间复杂度为 $O(|episode|YK)$。

4 实验设计和结果分析

本章利用仿真方法验证所提方法的有效性。参考文献[25-27]中的方法,基于 AGI STK(System Tool Kit)软件平台进行仿真验证。首先设置一个基于铱星星座的卫星簇,然后将用户随机放置在卫星簇的覆盖范围下,最后对卫星轨道参数进行设置,以更接近实际场景。将 STK 场景的时间跨度设置为一个小时,在这个时间段内基于$\mathcal{T}=600$ 和时间间隔 $\tau=1$ 获取卫星的轨迹数据样本。

在所提基于 DDRL 的框架下,DDQN 训练网络 Q 包含一个有 256 个神经元的隐藏层。TD3 训练网络 A, C_1 和 C_2 也各有一个包含 256 个神经元的隐藏层。DDQN 和 TD3 的目标网络 $\hat{Q}$, $\hat{A}$, $\hat{C}_1$ 和 $\hat{C}_2$ 与它们相应的训练网络具有相同的结构。在算法 2 中,设置采样回合数 $g_1=g_2=4$,更新回合数 $G_1=1000$, $G_2=100$。此外,回放缓冲区 $B_1=12000$, $B_2=60000$,总回合数 $episode=4000$。系统环境启动后,DDQN 首先进行学习,当它的缓冲区收集到 12000 个样本后,TD3 将开始学习,因为 TD3 需要更多的样本来稳定收敛。DDRL 使用Adam优化器进行优化,初始学习率为 0.00015。在1000回合后,学习率会下降 90%。此外,DDQN 和 TD3 的衰减因子分别设置为 0.9 和 0.99。ε 值的初始值设为 1.0,并线性下降至 0.02。TD3 使用正态分布来探索,且每次更新后,正态分布的方差将乘以 0.99。

此外,模拟环境中总带宽 B 的大小设置为 300 MHz,用户和卫星的发射功率 p_u 和 $p_s^{\wedge}$ 分别为 2 W 和 5 W[28]。成员卫星 s 所拥有的最大计算资源卫星 $f_s^{(\max)}$ 为 2～4 GHz[5]。卫星间的距离和通信能力分别为 800～1200 km 和 10 Gbps。此外,目标函数中的权重参数 ξ 均设置为 0.6。表 2 列出了仿真中的主要参数设置。

为了客观地评估性能,选取 3 种代表性的任务卸载和资源管理策略用于对比。

基准方法-1:DQN-DDPG 算法。考虑到本文设定场景是同时包含离散和连续动作空间的复杂环境,而深度 Q 网络(Deep Q-Network, DQN)和深度确定策略梯度(Deep Deter-

ministic Policy Gradient，DDPG）算法分别是作用于离散动作空间和连续动作空间中经典的深度强化学习（Deep Reinforcement Learning，DRL）算法，所以将两种算法结合，与本文方案进行对比。

基准方法-2：基于贪婪卸载策略的启发式算法[29]。该策略遵循贪心算法的原则，在每一步只选择当前看起来最优的决策，而不考虑整体最优解。

基准方法-3：随机方法。任务被随机卸载到一颗成员卫星上，并进行资源的随机分配，在这个过程中不考虑是否满足延迟和资源约束。

表 2　仿真参数

Table 2　Simulation parameters

参数名	数值
用户数量(U)	30
成员卫星数量(S)	9
系统时间(Y)	600
系统时间的时隙长度(τ)	1
任务到达时间间隔的平均值(μ)	1
任务数据大小($l_{u,t}$)	[0.1,1] Mbits
任务计算负载($c_{u,t}$)	[1,1.5]/(Kc/bits)
任务最大可容忍时延($T_{u,t}^{(\max)}$)	[5,10] s
卫星拥有的最大计算资源($f_s^{(\max)}$)	[2,4] GHz
有效电容系数(κ)	10^{-28}
用户发射功率(p_u)	2 W
簇首卫星发射功率(p)	5 W
噪声方差(σ^2)	7.9×10^{-16}
路径损耗指数(β)	2
簇首卫星与用户之间的总带宽(B)	300 MHz
ISL 通信能力(ω)	10 Gbps
卫星之间的距离($\varphi_{s,t}$)	[800,1 200] km

4.1　收敛性分析

本节中，通过计算平均累计奖励来评估 4 种算法的收敛性能。从图 5 中可以观察到，所提方法与基准方法-1 在最后的训练过程中都实现了稳定的任务卸载与资源分配过程。与基准算法-1 相比，所提方法获得了更优的累计奖励值。这是因为所提方法中的 DDQN 和 TD3 算法分别优化了 DQN 和 DDPG 算法中存在的过估计问题，可以减少学习过程中的振荡和不稳定性。此外，基准算法-2 和基准算法-3 不受训练过程的影响，性能较差，但是基准算法-2 在特定情况下可以找到最优解，性能相对基准算法-3 较好。

值得注意的是，所提方法在回合数较少时，其奖励值低于基准算法-2。这是因为所提方法在初期训练阶段需要探索环境，此时所提方法中的 DDQN 和 TD3 尚未充分学习到环境的最优策略，导致其奖励值较低。相比之下，基准方法-2 在初期更容易找到次优解，因为它只关注当前最优决策而不考虑长期收益。随着训练的进行，所提方法逐渐收敛到更优的策略，最终在后期表现出更高的奖励值。

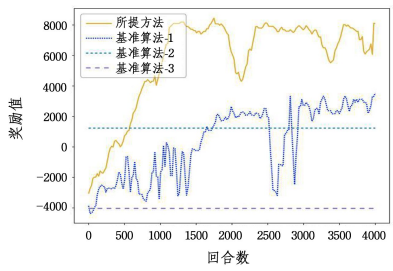


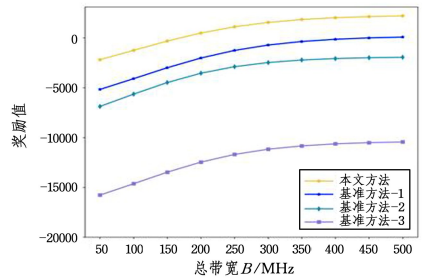
图 5　不同算法的收敛曲线

Fig. 5　Convergence curves of different algorithms

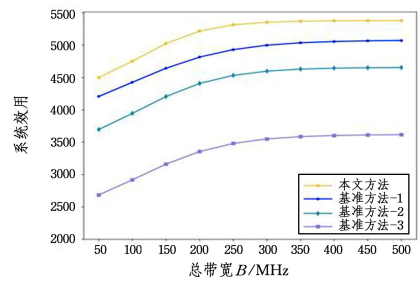
4.2　带宽和计算资源对性能的影响

首先，图 6 显示了当卫星具有不同的总带宽 B 时，每种策略的性能。从图 6(a)和图 6(c)可以观察到，当总带宽 B 较小时，每种算法的奖励率和任务成功率都很低。这是因为所有航空器用户统一接入簇首卫星并平均占用带宽，皆以较小的通信速率进行数据传输，增加了传输时延和能耗。随着 B 的逐渐增加，任务数据可以通过在通信资源更丰富的通道上进行传输，保证更多的任务按时完成并增加系统效用，如图 6(b)和图 6(c)所示。当 B 进一步增加时，大多数任务都可以成功完成，但是对系统效用的影响逐渐减小，此时每种算法的奖励都开始趋于稳定。此外，与 3 种基准算法相比，所提方法的性能最优，这意味着基于 DDRL 的任务卸载和资源分配策略在动态的通信带宽范围内都能找到最佳的卸载卫星和最优的资源分配策略。

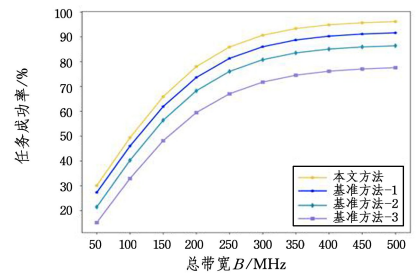
随着带宽 B 的增加，各项指标性能趋于平稳。当带宽 B 增加到一定程度时，通信资源的增加对任务传输时延的改善作用逐渐减弱，其不再是系统性能的瓶颈。此时，任务卸载和资源分配的性能主要受限于计算资源或其他因素。强化学习能够通过分析奖励函数和约束条件，揭示系统性能的上界，并在仿真中验证其接近或达到该上界的能力。



(a)奖励值



(b)系统效用



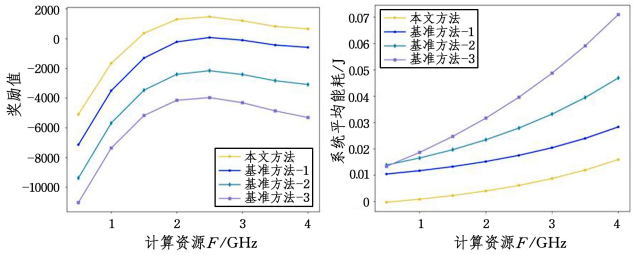
(c)任务成功率

图 6　不同算法在不同通信资源下的性能

Fig. 6　Performance of different algorithms under different communication resources

另外，比较了卫星具有不同计算能力 $f_s^{(\max)}$ 时联合策略

的变化，如图 7 所示。在图 7(a)中，当 $f_s^{(\max)}$ 很小(资源稀缺)时，任务无法及时分配到合适的计算资源进行处理，导致部分任务未能在规定时间内完成，从而使得强化学习过程中的惩罚值显著增大。随着 $f_s^{(\max)}$ 的增长，有更多的任务可以被成功处理，同时奖励也会正向增加。然而，当 $f_s^{(\max)}$ 在一定范围内进一步增加时，因为系统对 $f_s^{(\max)}$ 的需求已经得到了基本满足，所以此时不会显著提高任务成功率。相反，持续增加的 $f_s^{(\max)}$ 值会导致更大的计算能耗并呈指数级增长，最终导致奖励值在后半部分减少。从图 7(b)和图 7(c)可以看到，同样地，计算资源的增加会带来能耗的快速上升，导致系统能耗增加，但系统时延会因此降低。同时，所提方法仍保持着最好的性能。此外，从图 7(d)可以看到，每个算法任务的成功率都随着 $f_s^{(\max)}$ 的增加而增加，因为计算能力的提升可以带来处理时延的下降，从而提高任务成功率。其中，本文提出的 DDRL 方法保持了奖励最大化，而基准算法-1、基准算法-2 和基准算法-3 的性能依次降低。



(a)奖励值 (b)系统平均能耗

(c)系统平均时延 (d)任务成功率

图 7 不同算法在不同计算资源下的性能

Fig. 7 Performance of different algorithms under different computing resources

在低轨卫星网络中，通信带宽和计算资源的分配是相互依赖的。虽然通信资源的增加有助于缩短任务传输时延，但如果计算资源不足，任务最终无法被及时处理，对于通信资源而言属于一种浪费。类似地，计算资源的增加可以缩短任务处理时延，但如果通信资源不足，任务数据无法及时传输到计算节点。因此，通信与计算资源的协同优化是提升系统性能的关键。通过强化学习机制，所提方法能够在动态环境中实现通信与计算资源的最佳协同。

4.3 消融实验

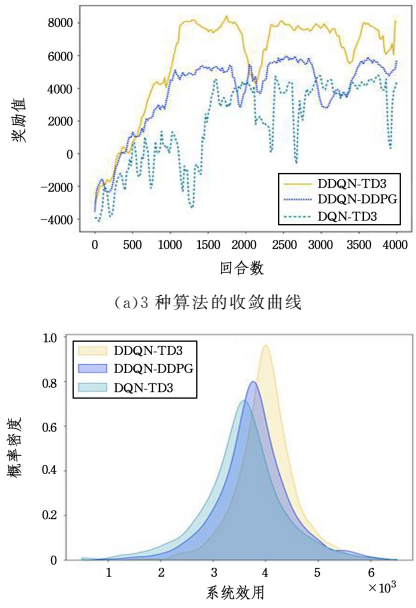
为了验证 DDRL 的有效性，接下来用两个修改的联合优化算法进行消融实验。

1)DDQN-DDPG 算法：在资源分配决策中使用 DDPG 算法替代 TD3 算法。DDPG 算法不使用双 Critic 架构，且不采用策略延迟更新的方法。

2)DQN-TD3 算法：在任务卸载决策中使用 DQN 算法替代 DDQN 算法。具体来说，使用同一个网络进行动作的选择与评估，而不使用单独的目标网络进行动作价值的评估。

如图 8(a)所示，DDQN-DDPG 和 DQN-TD3 收敛后的平均累计奖励分别为 5 000 和 4 000，均低于 DDQN-TD3 的 7 000。这是因为在处理需要高度精确价值估计的任务时，DDQN 和 TD3 通常比 DQN 和 DDPG 具有更高的性能和更快的收敛速度。此外，DDQN-DDPG 的平均累计奖励略高于 DQN-TD3，这是因为连续动作空间算法的输入需要离散动作空间算法的输出作为前提，而 DDQN 相对于 DQN 具有较好的性能和较高的鲁棒性。

图 8(b)展示了 3 种算法的系统效用分布概率密度，其中 DQN-TD3，DDQN-DDPG 和 DDQN-TD3 的系统效用分别集中在 3 600，3 800 和 4 100 附近，并且 DDQN-TD3 的曲线宽度更窄。3 条曲线对应的数学期望和方差说明 DDQN-TD3 具有最优的联合优化性能。



(a)3 种算法的收敛曲线

(b)系统效用概率密度

图 8 收敛曲线和概率密度

Fig. 8 Convergence curve and probability density

结束语 本文针对低轨卫星边缘计算的任务卸载和资源分配，提出了一种基于双深度学习算法的 DDRL 框架。该框架包含两个 DRL 算法，用来解决在任务卸载和资源分配过程中航空器用户通信及计算资源有限的问题。通过对所提方案的收敛性和不同资源下的性能进行分析，证明了所提方案的有效性。最后利用消融实验，证明了所提方案相比于其他 DRL 的组合具有更好的系统效用。下一阶段可研究将一个任务分割成一定数量的小任务，然后卸载到不同的卫星上进行分布式处理。

参 考 文 献

[1] MAO Y, YOU C, ZHANG J, et al. A survey on mobile edge computing: The communication perspective[J]. IEEE communications surveys & tutorials, 2017, 19(4): 2322-2358.

[2] DENBY B, LUCIA B. Orbital edge computing: Machine inference in space[J]. IEEE Computer Architecture Letters, 2019,

18(1):59-62.

[3] WEI J, HAN J, CAO S. Satellite IoT edge intelligent computing: A research on architecture[J]. Electronics, 2019, 8(11): 1247.

[4] CAO S, ZHAO Y, WEI J, et al. Space-based cloud-fog computing architecture and its applications[C]// 2019 IEEE World Congress on Services(SERVICES). IEEE, 2019: 166-171.

[5] GUO J, DU Y. Fog service in space information network: Architecture, use case, security and challenges[J]. IEEE Access, 2020, 8: 11104-11115.

[6] WEI J, CAO S. Application of edge intelligent computing in satellite Internet of Things[C]// 2019 IEEE International Conference on Smart Internet of Things(SmartIoT). IEEE, 2019: 85-91.

[7] WANG Y, YANG J, GUO X, et al. Satellite edge computing for the internet of things in aerospace[J]. Sensors, 2019, 19(20): 4375.

[8] XIE R, TANG Q, WANG Q, et al. Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues[J]. IEEE Network, 2020, 34(3): 224-231.

[9] LI C, ZHANG Y, HAO X, et al. Jointly optimized request dispatching and service placement for MEC in LEO network[J]. China Communications, 2020, 17(8): 199-208.

[10] CAO B, ZHANG J, LIU X, et al. Edge-cloud resource scheduling in space-air-ground-integrated networks for internet of vehicles [J]. IEEE Internet of Things Journal, 2021, 9(8): 5765-5772.

[11] CUI G, LONG Y, XU L, et al. Joint offloading and resource allocation for satellite assisted vehicle-to-vehicle communication[J]. IEEE Systems Journal, 2020, 15(3): 3958-3969.

[12] WANG B, FENG T, HUANG D. A joint computation offloading and resource allocation strategy for LEO satellite edge computing system[C]// 2020 IEEE 20th International Conference on Communication Technology(ICCT). IEEE, 2020: 649-655.

[13] CUI G, LI X, XU L, et al. Latency and energy optimization for MEC enhanced SAT-IoT networks[J]. IEEE Access, 2020, 8: 55915-55926.

[14] TRAN T X, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2018, 68(1): 856-868.

[15] LI N, YUE C F, GUO H B, et al. Design of large scale satellite cluster domain control strategy[J]. Chinese Space Science and Technology, 2023, 43(1): 18-28.

[16] CASONI M, GRAZIA C A, KLAPEZ M, et al. Integration of satellite and LTE for disaster recovery[J]. IEEE Communications Magazine, 2015, 53(3): 47-53.

[17] WANG F, JIANG D, QI S, et al. Fine-grained resource management for edge computing satellite networks[C]// 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-6.

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

[19] TANG Q, FEI Z, 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): 9169176.

[20] CUI G, LONG Y, XU L, et al. Joint offloading and resource allocation for satellite assisted vehicle-to-vehicle communication[J]. IEEE Systems Journal, 2020, 15(3): 3958-3969.

[21] DE SANCTIS M, CIANCA E, ARANITI G, et al. Satellite communications supporting internet of remote things[J]. IEEE Internet of Things Journal, 2015, 3(1): 113-123.

[22] CAO X, YANG B, SHEN Y, et al. Edge-assisted multi-layer offloading optimization of LEO satellite-terrestrial integrated networks[J]. IEEE Journal on Selected Areas in Communications, 2022, 41(2): 381-398.

[23] SONI G, SHARMA M. Performance Evaluation of a Free Space Optical Link-Based Inter Satellite Link(ISL) across Low Earth Orbit(LEO)[C]// 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and Its Control(PARC). IEEE, 2022: 1-5.

[24] ZHANG H, LIU R, KAUSHIK A, et al. Satellite edge computing with collaborative computation offloading: An intelligent deep deterministic policy gradient approach[J]. IEEE Internet of Things Journal, 2023, 10(10): 9092-9107.

[25] GAO X, LIU R, KAUSHIK A. Virtual network function placement in satellite edge computing with a potential game approach [J]. IEEE Transactions on Network and Service Management, 2022, 19(2): 1243-1259.

[26] GAO X, LIU R, 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.

[27] LI Q, WANG S, MA X, et al. Service coverage for satellite edge computing[J]. IEEE Internet of Things Journal, 2021, 9(1): 695-705.

[28] CAO X, YANG B, SHEN Y, et al. Edge-assisted multi-layer offloading optimization of LEO satellite-terrestrial integrated networks[J]. IEEE Journal on Selected Areas in Communications, 2022, 41(2): 381-398.

[29] WU Y C, DINH T Q, FU Y, et al. A hybrid DQN and optimization approach for strategy and resource allocation in MEC networks[J]. IEEE Transactions on Wireless Communications, 2021, 20(7): 4282-4295.

LI Fang, born in 1999, postgraduate. Her main research interest is intelligent network computing.

SHEN Hang, born in 1984, Ph.D, associate professor, master's supervisor, is a senior member of CCF(No. 19088S). His main research interest is space-air-ground integrated networks.

（责任编辑：何杨）