基于深度确定性策略梯度的星地融合网络可拆分任务卸载算法

宋晓勤[1,2]，吴志豪[1]，赖海光[3]，雷磊[1]，张莉涓[1]，吕丹阳[1]，郑成辉[3]
（1. 南京航空航天大学电子信息工程学院，江苏 南京 210016；2. 东南大学移动通信全国重点实验室，江苏 南京 210096；
3. 南京控维通信科技有限公司，江苏 南京 211135）

摘　要：为解决低轨卫星网络中星地链路任务卸载时延长的问题，提出了一种基于深度确定性策略梯度（DDPG）的星地融合网络可拆分任务卸载算法。针对不同地区用户建立了星地融合网络的多接入边缘计算结构模型，通过应用多智能体DDPG算法，将系统总服务时延最小化的目标转化为智能体奖励收益最大化。在满足子任务卸载约束、服务时延约束等任务卸载约束条件下，优化用户任务拆分比例。仿真结果表明，所提算法在用户服务时延和受益用户数量等方面优于基线算法。

关键词：星地融合网络；深度确定性策略梯度；资源分配；多接入边缘计算
中图分类号：TN92
文献标志码：A
DOI: 10.11959/j.issn.1000–436x.2024181

Split task offloading algorithm for satellite-ground integrated networks based on deep deterministic policy gradient

SONG Xiaoqin[1,2], WU Zhihao[1], LAI Haiguang[3], LEI Lei[1],
ZHANG Lijuan[1], LYU Danyang[1], ZHENG Chenghui[3]
1. College of Electronics and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
2. National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
3. Nanjing Cowave Communication Technology Co.,Ltd., Nanjing 211135, China

Abstract: To address the prolonged task offloading delay in low earth orbit satellite networks, a split task offloading algorithm based on deep deterministic policy gradient (DDPG) was proposed for satellite-ground integrated networks. A multi-access edge computing structural model of the satellite-ground integrated network was established for users in different regions. By applying a multi-agent DDPG algorithm, the objective of minimizing total system service delay was transformed into maximizing agent reward returns. Under the constraints of sub-task offloading, service delay, and other task offloading conditions, the user task splitting ratio was optimized. Simulation results indicate that the proposed algorithm outperforms baseline algorithms in terms of user service delay and the number of benefited users.

Keywords: satellite-ground integrated network, deep deterministic policy gradient, resource allocation, multi-access edge computing

0　引言

随着星地融合网络的快速发展，频谱资源、功率分配和计算能力面临着日益严峻的挑战，尤其在低时延要求的场景下，资源的高效分配成为

收稿日期：2024–08–17；**修回日期：**2024–09–29
基金项目：国家自然科学基金资助项目（No.62371232）；东南大学移动通信全国重点实验室开放研究基金资助项目（No.2024D13）；未来网络科研基金资助项目（No.FNSRFP-2021-ZD-4）
Foundation Items: The National Natural Science Foundation of China (No.62371232), The Open Research Fund of National Mobile Communications Research Laboratory, Southeast University (No.2024D13), The Future Network Scientific Research Fund Project (No.FNSRFP-2021-ZD-4)

关键问题[1]。多接入边缘计算（MEC, multi-access edge computing）作为应对计算需求增长和降低时延的核心技术，将计算资源下沉至边缘节点，能有效缓解核心网络负载[2-3]。传统地面MEC节点受限于固定部署，难以满足大范围、动态变化的任务卸载需求。星地融合网络通过整合卫星与地面MEC资源，不仅扩展了广域覆盖，还能根据用户需求动态分配计算任务，提升服务质量[4-5]。但其资源分配面临两大挑战：一是低轨卫星（LEO, low earth orbit）的轨道高度和持续变化导致星地链路时延较高，并且引发网络拓扑的频繁变动；二是密集用户与稀疏用户场景的资源需求差异显著，传统算法效率低下且计算开销大。

随着深度学习和强化学习技术的不断突破，深度确定性策略梯度（DDPG, deep deterministic policy gradient）算法作为一种高效的资源分配方法，能够从复杂的环境状态中直接学习到最优的连续动作策略，特别适用于卫星网络中动态性强和时延高的条件，并且可以在密集用户和稀疏用户场景中灵活调整资源分配。训练完成后，该模型可在不需要频繁重新训练的情况下，根据当前环境状态实时做出决策，极大地提高了资源分配的效率和灵活性。

星地融合网络资源管理涵盖多个关键技术，现有研究主要集中在网络切片、中继调度和任务卸载等方面。网络切片通过将物理网络资源虚拟化成多个逻辑网络，能够针对不同服务需求进行资源隔离和动态管理。文献[6]提出了一种将网络切片应用于星地融合网络的资源管理系统，通过排序新用户的网络切片请求实现资源的高效分配。中继调度旨在优化卫星和地面基站之间的中继通信，依据卫星的可用服务时间来调度中继流量。文献[7]研究了星地融合网络的中继问题，利用泰勒级数近似的几何规划方法进行建模，并生成配置矩阵来优化中继调度。任务卸载通过将计算任务从设备端卸载到边缘服务器或云端，降低计算负载和功耗。文献[8]提出了一种基于学习的队列感知任务卸载和资源分配算法，通过拉格朗日对偶分解在设备端进行任务拆分，使用基于演员-评价者（AC, actor-critic）框架的算法应对维度灾难问题。文献[9]提出了一种速率分割多址的资源分配算法，通过优化速率和功率资源，显著提升了星地融合网络中的传输速率。

综上所述，现有研究在网络切片、中继调度和任务卸载等方面均取得了重要进展，形成了多个针对星地融合网络资源管理问题的解决方案，但在应对动态复杂环境和大规模数据时表现不足。相比之下，机器学习能够自主学习，快速适应网络变化，有效优化资源分配。文献[10]针对星地链路上行信道提出了一种基于深度强化学习（DRL, deep reinforcement learning）的在线信道分配和功率控制算法，以平衡提高资源效率和满足服务质量需求之间的性能。文献[11]针对计算任务不可微分的卸载问题提出了一种异步优势的DRL算法，大大加快了模型的收敛速度。文献[12]提出了一种基于无人机（UAV, unmanned aerial vehicle）中继的多智能体DRL算法，极大提高了UAV的能量利用效率和卫星的资源分配效率。然而，上述研究中的用户模型相对单一，较少涉及无地面网络覆盖的稀疏用户和流量密集区域的复杂场景，同时未充分考虑任务的可分解性属性，这限制了资源分配的效率和灵活性。

因此，提高星地融合网络用户的接入数量和网络体验，对缓解星地服务器的资源计算压力具有非常重要的理论意义和应用价值。本文主要的研究工作与贡献如下。

1) 针对地面网络密集和地面网络稀疏两类用户，建立星地融合网络多接入边缘计算结构模型。地面用户设备根据卸载策略将任务卸载到本地网络、地面网络和卫星网络，最小化并行处理服务时延中的最大值，并满足地面用户设备均完成所有子任务约束和所有用户的每个子任务服务时延小于最大时延门限值约束。

2) 提出了一种基于深度确定性策略梯度的星地融合网络可拆分任务卸载算法，命名为可拆分任务卸载-深度确定性策略梯度（STO-DDPG, split task offloading based on DDPG）算法，将系统的优化目标转换为深度强化学习网络的训练问题。以最小化系统服务时延为目标，在满足用户最大时延上限和多接入边缘计算服务器资源上限等约束条件下，利用系统模型的高度动态性不断调整网络参数，实现用户任务卸载的最优拆分比例。

3) 仿真结果表明，本文算法具有很好的收敛性和稳定性，同时具有较低的平均服务时延和良好的能效性能。与传统算法和全卸载算法对比，本文算法在诸多性能指标上都有明显的优势，验证了其有效性。

1 系统模型

1.1 星地融合网络模型

星地融合网络多接入边缘计算模型主要由卫星网络、地面网络和地面用户组成，如图1所示。该模型通过卫星网络弥补地面网络在偏远区域和高密度区域的覆盖不足与高时延问题。针对低轨卫星高度动态性和星地链路时延长的问题，应用深度强化学习算法实现智能化的资源动态调度，提高计算资源和通信资源的分配效率并降低服务时延，以适应网络的复杂变化。

卫星网络由运行在轨道上的多个低轨卫星组成，卫星数量共有M个，表示为集合$\mathcal{M}=\{1,2,3,\cdots,M\}$，每个低轨卫星均配备一个MEC服务器。卫星网络可以覆盖地面基础设施不足的偏远地区，这些MEC服务器能够在卫星所处的低轨道位置为附近的用户设备提供计算资源，以实现数据的处理、存储等功能。

地面网络主要由大面积分布的地面基站构成，每个地面基站配备有强大的天线，可以为大量的用户提供计算服务。地面基站的数量共有N个，表示为集合$\mathcal{N}=\{1,2,3,\cdots,N\}$，每个地面基站均配备一个MEC服务器，用于用户的任务卸载工作。地面用户主要是指星地融合网络中的用户设备，如智能手机、智能汽车、多媒体设备等。用户设备拥有一定量的本地计算资源，但通常计算能力较低[13]。

卫星MEC服务器具有一定量的计算资源，可以为地面用户提供相应的计算卸载服务，具体可以表示为

$$m=\{C_m,B_m,R_m,S_m,U_m\} \tag{1}$$

其中，C_m表示卫星MEC服务器m的总计算资源（以CPU周期为单位），B_m表示卫星信道带宽，R_m表示卫星与地面之间的链路传输速率，S_m表示卫星与用户之间的距离，这里用卫星的轨道高度来表示，U_m表示卸载在卫星MEC服务器m上的用户任务量形成的集合。

星地链路之间的链路传输速率R_m表示为[13]

$$R_m=B_m\mathrm{lb}\left(1+\frac{ph_m}{\sigma_m^2}\right) \tag{2}$$

其中，p表示用户传输功率，σ_m^2表示卫星信道中的高斯白噪声，h_m表示卫星信道的总传输增益，可以表示为

$$h_m=\left|H_m\right|^2G_mL_m \tag{3}$$

其中，H_m为卫星和用户之间的莱斯信道矩阵，G_m为天线增益，L_m为星地链路的传播损耗。

相对于卫星MEC服务器，地面MEC服务器与用户之间的距离更近，且整个服务的通信时延更低，因此地面MEC服务器承担着整个系统中绝大部分的任务卸载工作。地面MEC服务器可以表

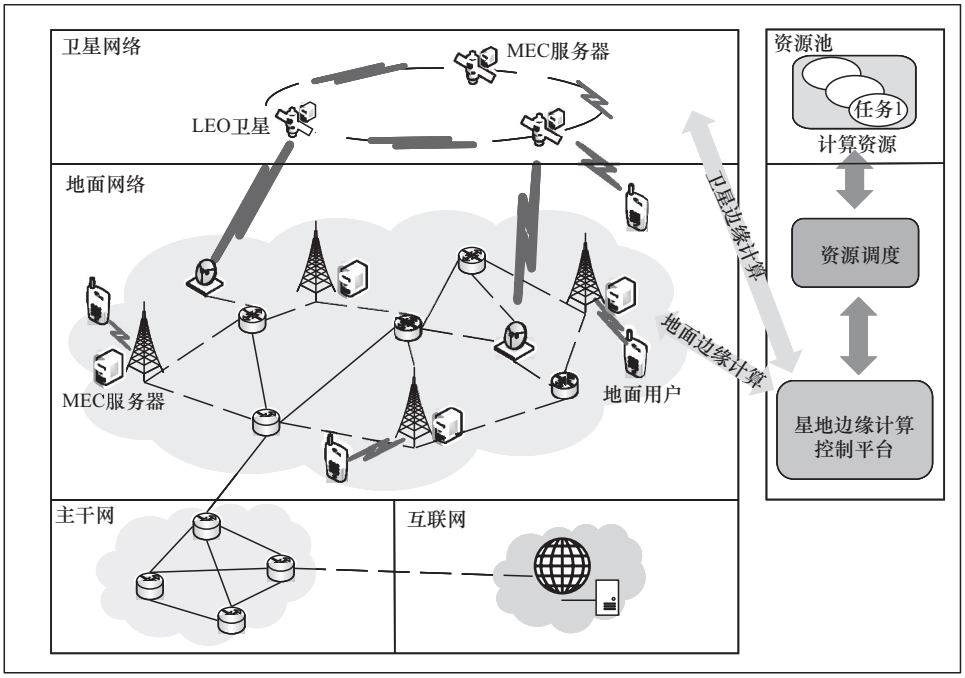

图1 星地融合网络多接入边缘计算模型

示为

$$n = \{C_n, B_n, R_n, S_n, U_n\} \tag{4}$$

其中，地面链路的传输速率R_n可以表示为

$$R_n = B_n \mathrm{lb}\left(1 + \frac{ph_n}{\sigma_n^2}\right) \tag{5}$$

其中，地面信道传输增益h_n通常与距离的幂次方成反比，可以表示为[14]

$$h_n = (\mathrm{dis})^{-\lambda} \tag{6}$$

其中，dis 表示用户与地面 MEC 服务器之间的距离，λ表示路径损耗因子。

可拆分任务是指能够被细分为多个独立或半独立的子任务并能够在不同计算节点上并行处理的计算任务，具备并行性、独立性和低耦合性。当计算任务需要大量的计算资源，超出了本地设备的处理能力，且可以被分解成多个并行处理的子任务时需要进行拆分卸载。假定每个地面用户都携带一个时延敏感型的计算任务，且该任务可被拆分成多个相互独立的子任务。在模型中假定每个用户均处在每一个 MEC 服务器的覆盖范围内，即每个用户的多个子任务可以卸载到不同的 MEC 服务器中进行处理。用户集合可以用$\mathcal{I} = \{1,2,3,\cdots,I\}$表示，每个用户模型可表示为

$$i = \{X_i, C_i, D_i, Z_i\} \tag{7}$$

其中，X_i表示用户 i 的总计算任务量，C_i表示用户 i 本地的计算资源量，D_i表示用户 i 的总计算任务传输数据量，Z_i表示用户 i 的子任务分配决策集，即用户 i 分配到自身及每个 MEC 服务器的计算任务比例，可以表示为

$$Z_i = \{z_i^0, z_i^1, \cdots, z_i^{M+N}\} \tag{8}$$

其中，$z_i^0 \in [0,1]$表示用户 i 本地处理的任务比例，其他值为相应卸载到 MEC 服务器的任务比例。对于每个Z_i，均满足式(9)。

$$\sum_{k=0}^{M+N} z_i^k = 1,\ \forall i \in \mathcal{I} \tag{9}$$

即用户总任务卸载比例之和为 1。

系统控制平台用于整合系统的计算资源，根据用户的任务卸载比例分配计算资源，并返回任务卸载结果，优化最佳卸载比例。

1.2　多接入边缘计算优化问题设计

虽然多接入边缘计算能够极大地减少服务时延，然而时延或时延抖动仍是难以避免的，因此设计更加合理的计算卸载策略用于减少服务时延是十分有必要的。根据地面用户所处位置不同，采用对应的计算卸载策略可以有效减少服务时延。

1) 本地任务处理

对于卸载到用户本地处理的子任务，整个服务时延只包含子任务处理时延，具体可以表示为

$$t_i^0 = \frac{z_i^0 X_i}{C_i} \tag{10}$$

2) 地面 MEC 服务器卸载

对于卸载到地面网络中的子任务，整个服务时延包括计算时延和传输时延两部分。与大部分研究所讨论的一样，由于地面终端在完成计算任务后，返回链路中的数据量通常非常小，如对象识别和确认信息的结果，仅包含极少的比特数，因此可忽略下行传输带来的时延[15]。此时用户 i 卸载至第 n 个地面 MEC 服务器的服务时延可以表示为

$$t_i^n = \frac{z_i^n X_i}{C_n^i} + \frac{z_i^n D_i}{R_n^i} \tag{11}$$

其中，C_n^i为地面 MEC 服务器 n 给用户 i 分配的计算资源量，R_n^i为地面 MEC 服务器 n 到用户 i 的传输速率。为了避免任务拆分后卸载到地面 MEC 服务器的子任务计算量可能存在的较大差异，本文将按照计算量的比例为不同的子任务分配计算资源，具体表示为

$$C_n^i = \frac{z_i^n X_i}{\sum_{i=1}^{I} U_n\{z_i^n X_i\}} C_n \tag{12}$$

3) 卫星 MEC 服务器卸载

当地面用户向卫星 MEC 服务器卸载子任务时，星地链路之间的长传输时延是无法避免的，同时每个卫星中因为嵌入了一个 MEC 服务器，会相应地增加卫星的建设成本和能量消耗，不过得益于卫星的大面积覆盖能力，此时所有地面用户可以连接至卫星，从卫星 MEC 卸载服务中受益。对于卸载至卫星网络中的子任务，整个服务时延包括计算、传输和传播时延 3 个部分，且同样不考虑下行传输时延。此时用户 i 卸载至第 m 个卫星 MEC 服务器的服务时延可以表示为

$$t_i^m = \frac{z_i^m X_i}{C_m^i} + \frac{z_i^m D_i}{R_m^i} + \frac{2S_m^i}{c} \tag{13}$$

其中，$\frac{S_m^i}{c}$为链路传播时延，c 为光速，C_m^i为卫星 MEC 服务器 m 给用户 i 分配的计算资源，R_m^i为卫星 MEC 服务器 m 到用户 i 的传输速率。

地面 MEC 服务器在用户与服务器距离较近时

具有较低的传输时延，适用于网络基础设施较好的城市或局部区域内的任务卸载。而卫星MEC则适合用于地面网络覆盖不足的地区，或在大量用户同时请求服务时提供补充计算资源。在实际部署中，可以根据用户所处的地理位置、网络覆盖情况及任务的时延敏感性，灵活选择使用地面MEC或卫星MEC，也可以采取混合卸载策略。优先考虑地面MEC，若地面网络资源紧张或不可用，则将部分任务卸载至卫星MEC，从而优化整体的网络性能和用户体验。

由于每个用户的多个子任务可以并行地在本地或MEC服务器中处理，因此用户i的总服务时延T_i取决于多个并行处理服务时延中的最大值，即

$$T_i\left(Z_i,Z_{-i}\right)=\max\left\{t_i^0,t_i^n,t_i^m\right\},\ n\in\mathcal{N},m\in\mathcal{M} \tag{14}$$

其中，Z_{-i}表示除用户i之外的其他所有用户的子任务分配决策集合。

用户开销即用户整个卸载任务周期内的能耗，基于上述模型描述，每个用户的开销主要由以下三部分构成。

① 本地计算开销。对于在本地处理的子任务，依据文献[16]中所述，用户处理计算任务时的开销与其自身的计算能力以及处理的任务CPU周期总数的平方成正比。因此，用户的本地计算开销可以表示为

$$\mathrm{sp}_i^0=\delta z_i^0X_i(C_i)^2 \tag{15}$$

其中，δ为一个常量，这里取10^{-27}。

② 子任务传输开销。对于卸载到MEC的子任务，用户需要以恒定的功率传输子任务，直到任务传输结束，同样因计算结果数据量远小于任务本身数据量，因此下行传输开销可以忽略不计。

当用户i选择卸载至地面MEC服务器n时的子任务传输开销表示为

$$\mathrm{sp}_i^n=p\frac{z_i^nD_i}{R_n^i} \tag{16}$$

用户i选择卸载至卫星MEC服务器m的子任务传输开销表示为

$$\mathrm{sp}_i^m=p\left(\frac{z_i^mD_i}{R_m^i}+\frac{S_m^i}{c}\right) \tag{17}$$

③ 子任务处理等待开销。当卸载至MEC的子任务传输结束时，用户需要以p_t的待机功率等待MEC任务处理结束，此时的用户开销即为等待开销。用户i的等待开销值表示为

$$\mathrm{sp}_i^{\mathrm{wait}}=\begin{cases}p_t\left(\dfrac{z_i^nX_i}{C_n^i}\right),\ n\in\mathcal{N}\\ p_t\left(\dfrac{z_i^mX_i}{C_m^i}\right),\ m\in\mathcal{M}\end{cases} \tag{18}$$

因此，用户i的总开销sp_i为本地计算开销、子任务传输开销和子任务处理等待开销之和，具体表示为

$$\mathrm{sp}_i=\mathrm{sp}_i^0+\sum_{n=1}^{N}\mathrm{sp}_i^n+\sum_{m=1}^{M}\mathrm{sp}_i^m+\sum_{\mathrm{wait}=1}^{M+N}\mathrm{sp}_i^{\mathrm{wait}} \tag{19}$$

整个系统的总开销SP为所有用户总开销之和，具体表示为

$$\mathrm{SP}=\sum_{i=1}^{I}\mathrm{sp}_i \tag{20}$$

综上所述，对于用户采用的可拆分任务卸载模式，在上述条件下，由于整个系统的计算资源是共享的，因此用户任务分配决策之间会相互影响。多接入边缘计算的优化问题表示为

$$\begin{aligned}&\min\sum_i^IT_i\\&\text{s.t. C}_1\text{: sum}\left(Z_i\right)=1,\forall i\in\mathcal{I}\\&\text{C}_2\text{:}t_i^j\leqslant T_i^{\mathrm{threshold}},\forall i\in\mathcal{I},\forall j\in\{0\}\cup\mathcal{N}\cup\mathcal{M}\end{aligned} \tag{21}$$

其中，优化目标为最小化整个网络总服务时延；约束条件C_1保证每个用户无论通过本地处理或MEC卸载处理其子任务分配决策之和为1，由任务卸载比例总和式(9)确定；约束条件C_2表示所有用户的每个子任务服务时延均应小于预先设定的最大时延门限值，从而确保每个用户最终的服务时延满足要求，由算法奖励式(34)确定。

2 算法设计

2.1 基于多接入边缘计算的DDPG算法模型

DDPG算法作为一种使用深度神经网络的无模型AC策略算法，可以学习连续动作空间中的策略。在多智能体的环境下，可以采用集中式训练和分布式执行方式处理。在集中式训练过程中，每个智能体的Critic都具备对全局环境信息的访问权限，这样可以对整个系统状态和所有智能体的策略进行综合评估，并优化其行为。一旦集中式训练完成，各智能体的Actor便能在分布式执行阶段独立做出行动决策，不再需要全局信息，增强了系统的隐私性和可扩展性。

DDPG算法共有4个深度神经网络，分别是策略函数网络$\mu(s;\theta^u)$、Q值函数网络$Q(s,a;\theta^Q)$以

及二者对应的具有相同结构的目标网络，即具有参数$\theta^{\mu'}$的策略函数目标网络μ'和具有参数$\theta^{Q'}$的Q值函数目标网络Q'。

DDPG的损失函数一般可以表示为

$$L\left(\theta^{Q}\right)=E_{\mu'}\left[\left(y_k-Q\left(s_k,a_k;\theta^{Q}\right)\right)^2\right] \tag{22}$$

其中，y_k为目标函数，具体表示为

$$y_k=r_k+\gamma Q\left(s_{k+1},\mu\left(s_{k+1}\right);\theta^{Q}\right) \tag{23}$$

DDPG的策略梯度可以通过链式规则更新，表示为

$$\nabla_{\theta^{\mu}}J\approx E_{\mu'}\left[\nabla_{\theta^{\mu}}Q\left(s,a;\theta^{Q}\right)|_{\ s=s_k,a=\mu\left(s_k;\theta^{\mu}\right)}\right]=$$

$$E_{\mu'}\left[\nabla_a Q\left(s,a;\theta^{Q}\right)\Big|_{s=s_k,a=\mu\left(s_k;\theta^{\mu}\right)}\nabla_{\theta^{\mu}}\mu\left(s;\theta^{\mu}\right)\Big|_{s=s_k}\right] \tag{24}$$

为了能够更充分地探索状态空间，需要平衡探索和开发之间的权衡，由于DDPG是一种非策略算法，可以独立于学习过程来处理DDPG的探索，因此可通过添加行为噪声n_k来构造动作空间，进而获得最终的动作策略。

$$a_k=\mu\left(s_k\right)+n_k \tag{25}$$

行为噪声n_k服从高斯分布。

$$n_k\sim\mathbb{N}\left(\mu_e,\sigma_{e,k}^2\right) \tag{26}$$

其中，μ_e表示均值，$\sigma_{e,k}^2$表示方差。

将策略网络作为智能体，在环境中执行完动作a_k后，观察下一个状态s_{k+1}和即时回报奖励r_k，同时将训练经验(s_k,a_k,r_k,s_{k+1})存储在回访缓冲区中，然后智能体选择几个跃迁经验组成小批量，并将其输入策略函数网络和Q值函数网络。通过这些小批量跃迁经验，策略函数目标网络μ'输出行为值$\mu'(s_{k+1})$到Q值函数目标网络Q'，Q值函数网络Q可以计算目标值[17]。

为了最小化损失函数，Q值函数网络Q由给定的优化器更新，然后策略函数网络将小批量动作$a=\mu(s_k)$给Q值函数网络实现动作a的梯度，参数可以由其自身的优化器导出。使用这2个梯度，可以将策略函数[18]网络更新为

$$\nabla_{\theta^{\mu}}J\approx$$

$$\frac{1}{N}\sum_j\left[\nabla_a Q\left(s,a;\theta^{Q}\right)\Big|_{s=s_k,a=\mu\left(s_k;\theta^{\mu}\right)}\nabla_{\theta^{\mu}}\mu\left(s;\theta^{\mu}\right)\Big|_{s=s_k}\right] \tag{27}$$

最后，DDPG通过一个软更新常数τ来更新策略函数目标网络和Q值函数目标网络，可以分别表示为

$$\begin{aligned}\theta^{Q'}&\leftarrow\theta^{Q}+(1-\tau)\theta^{Q'}\\ \theta^{\mu'}&\leftarrow\theta^{\mu}+(1-\tau)\theta^{\mu'}\end{aligned} \tag{28}$$

由于上述网络模型中存在连续的策略取值，因此本节通过DDPG深度强化学习算法对网络模型进行训练优化，最终执行训练好的网络模型得出最佳的任务卸载策略。

图2给出了基于DDPG的多接入边缘计算策略模型优化示意图。基于图2，对DDPG算法模型给出如下详细说明。

1) 状态s。状态s用于表征整个系统当前时刻的具体情况，对于本文的星地融合网络多接入边缘计算模型中的状态可以表示为[19]

$$s=\begin{bmatrix}C_1^1 & C_1^2 & \cdots & C_1^I\\ C_2^1 & C_2^2 & \cdots & C_2^I\\ \vdots & \vdots & \ddots & \vdots\\ C_K^1 & C_K^2 & \cdots & C_K^I\end{bmatrix}_{(K=M+N)I} \tag{29}$$

以每个MEC服务器中每个用户的计算资源分配矩阵作为状态s，当用户i未向地面MEC服务器n卸载子任务时，其计算资源分配量为0，即$C_n^i=0$。因此，该矩阵可以同时表征星地融合网络中卫星链路和地面基站之间资源调度和任务分配的全局情况。

2) 动作a。动作a为所有用户子任务分配策略的集合。对动作a的定义如式(30)所示。

$$a=\left\{\mathrm{Pro}_{\mathcal{I}(\mathcal{M}+\mathcal{N}+1)}\right\} \tag{30}$$

其中，$\mathrm{Pro}_{\mathcal{I}(\mathcal{M}+\mathcal{N}+1)}$表示所有用户任务卸载分配集合，具体表示为

$$\mathrm{Pro}_{\mathcal{I}(\mathcal{M}+\mathcal{N}+1)}=\{Z_1,Z_2,\cdots,Z_i,\cdots,Z_I\} \tag{31}$$

其中，Z_i表示第i个用户的卸载策略，其具体卸载包括本地卸载、地面MEC卸载和卫星MEC卸载，如式(8)所示。为了能够尽可能地对环境进行探索，同时也要使算法最后能够稳定地收敛，对动作a添加一个逐次递减的干扰噪声，干扰噪声服从式(32)所示的正态分布。其中，噪声的方差σ是一个随迭代递减的值，递减值为σ_t。

$$f(x)=\frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{32}$$

其中，μ为正态分布的均值，取值为未添加干扰噪声之前的动作a。对添加干扰之后的动作做归一化处理，得到最终的动作a，表示为

$$a=\left\{\frac{Z_1}{\mathrm{sum}\,(Z_1)},\frac{Z_2}{\mathrm{sum}\,(Z_2)},\cdots,\frac{Z_I}{\mathrm{sum}\,(Z_I)}\right\} \tag{33}$$

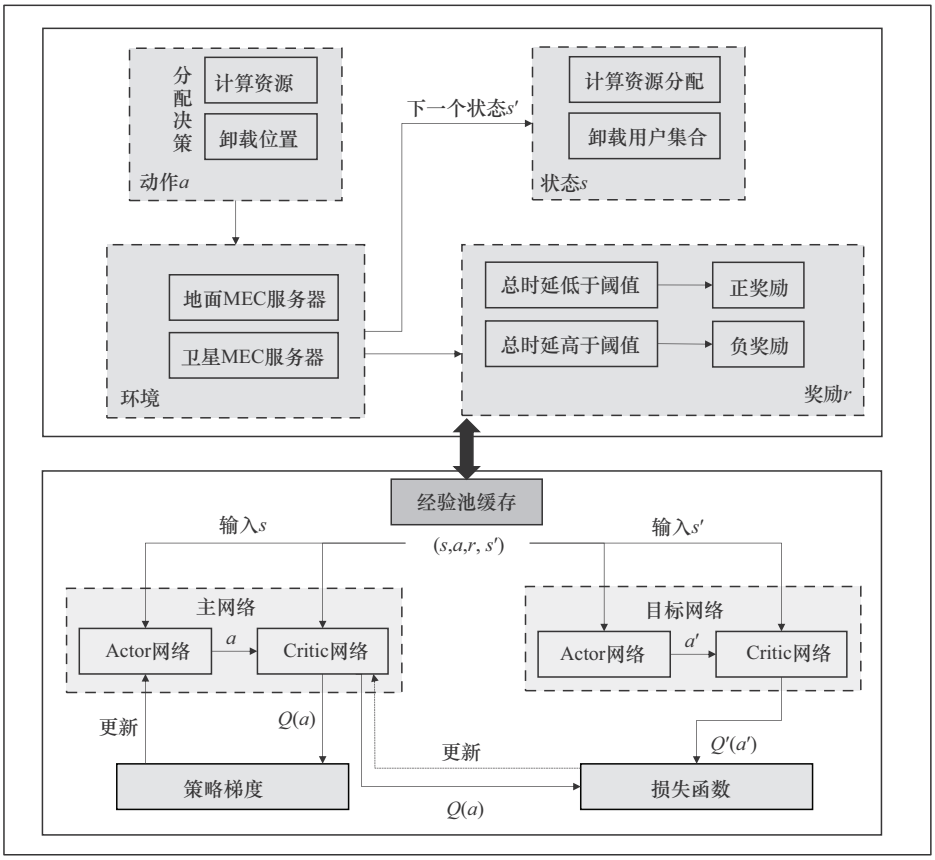

图2 基于DDPG的多接入边缘计算策略模型优化示意

3) 奖励r。奖励收益作为评估动作好坏的指标，很大程度上影响着模型训练网络的性能。对上述模型中的奖励定义如式(34)所示。

$$r = T^{\text{th}} - \sum_{i}^{I} T_i \tag{34}$$

其中，T_i表示第i个用户并行处理中的最大服务时延，见式(14)，T^{th}表示模型训练设定的时延阈值。从式(34)中可以看出，当总时延低于阈值时，奖励收益为正奖励，且总时延越低，奖励越大；当总时延高于阈值时，奖励收益为负奖励，且总时延越高，惩罚越大，以达到最小化总服务时延的目标。

综上所述，通过DDPG算法，模型会不断训练学习，在不同的环境状态下，采取各种动作与环境作用获得不同的奖励，最终朝最大的奖励，即最小系统总服务时延的方向收敛。

2.2 多接入边缘计算资源分配管理

本节对整个模型中的多接入边缘计算资源分配管理流程展开说明。

1) 用户生成计算任务。任务中包含该任务的总数据量D_i及处理整个任务所需的计算资源量X_i。

2) 用户子任务切分。用户初始化选择子任务卸载位置及相应的子任务卸载比例。对密集人群用户来说，子任务卸载位置可以选择地面MEC和卫星MEC，且地面MEC整体时延比较小。而稀疏人群用户只能选择卫星MEC，且整体时延要高于密集人群用户。每个用户的子任务卸载比例在满足总比例之和为1的前提下，由系统控制平台计算每个子任务的卸载服务时延，所有子任务时延均需要小于预设的时延门限值。

3) 服务时延计算。依据2)中的子任务卸载比例，分别代入式(10)、式(11)和式(13)中计算本地处理、地面MEC卸载和卫星MEC卸载中所有子任务的服务时延，因为任务采取并行处理方式，通过式(14)计算所有子任务服务时延中最大的值作为用户的最终服务时延，并更新计算每个MEC服务器中用户的子任务卸载情况及计算资源的分配情况。

4) DDPG深度神经网络构建。以3)中的用户子任务卸载情况和MEC服务器计算资源的分配情况作为深度神经网络的状态输入，计算出2)中用户需

要的子任务卸载决策比例作为动作输出，将动作输入系统环境中作用，并计算出该动作决策下的系统总服务时延，代入式(34)得到动作的奖励回报值，同时计算出下一步状态。将每一对状态、动作、奖励回报和下一步状态输入深度神经网络经验池中用来训练网络。

5) 集中式训练。取深度神经网络经验池中的小批量数据训练网络，当系统总服务时延稳定在最低点附近或者算法总奖励回报值不再上升时，整个网络的性能将不会再继续优化提升，此时算法收敛，保存网络参数。基于DDPG的可拆分任务卸载算法网络训练伪代码如算法1所示。

算法1　STO-DDPG算法之集中式训练

初始化网络系统的属性参数：计算资源、分配策略等。随机初始化Critic网络$Q\left(s,a,\theta^{Q}\right)$和Actor网络$\mu\left(s,\theta^{\mu}\right)$以及Critic网络和Actor网络的参数。初始化目标网络Q'和μ'。初始化回放经验池Buffer，设定软更新学习率τ和折扣因子γ

① for each episode do
② 初始化随机噪声
③ 初始化起始状态s_0
④ for each step do
⑤ 选择行动值并添加随机噪声并进行归一化处理得到动作a
⑥ 依据策略网络卸载子任务，并分配计算资源，得到回报奖励r及下一步状态s'
⑦ 存储(s,a,r,s')到经验池Buffer中，从经验池中抽出一个batch的数据进行训练
⑧ 依据目标网络计算
$y = r + \gamma Q'(s',\mu'(s',\theta^{\mu'});\ \theta^{Q'})$
⑨ 以最小化Loss值更新Critic网络
$$\text{Loss} = \frac{\sum_{k=1}^{K}(y_k - Q(s_k,a_k,\theta^{Q}))^2}{N}$$
⑩ 以采样策略梯度更新策略
⑪ 更新目标网络$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$和$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$
⑫ end for
⑬ end for
⑭ 算法收敛

6) 分布式执行。执行5)中训练好的网络模型，由于该系统模型是高度动态的，因此采用一边执行一边训练的方式，即不断调整网络参数，得出最优的子任务分配策略，网络执行伪代码如算法2所示。

算法2　STO-DDPG算法之分布式执行

初始化网络系统的属性参数：计算资源、分配策略等。初始化回放经验池Buffer，设定软更新学习率τ和折扣因子γ

① 用网络模型
② for each episode do
③ 初始化起始状态s_0
④ for each step
⑤ 依据策略网络卸载子任务并分配计算资源，得到回报奖励r及下一步状态s'
⑥ 存储(s,a,r,s')到经验池Buffer中，从经验池中抽出一个batch的数据进行训练
⑦ 依据目标网络计算
$y = r + \gamma Q'(s',\mu'(s',\theta^{\mu'});\ \theta^{Q'})$
⑧ 以最小化Loss值更新Critic网络
$$\text{Loss} = \frac{\sum_{k=1}^{K}(y_k - Q(s_k,a_k,\theta^{Q}))^2}{N}$$
⑨ 以采样策略梯度更新策略
⑩ 更新目标网络$\theta^{Q'}$和$\theta^{\mu'}$
⑪ end for
⑫ end for
⑬ 输出最终子任务分配策略
$$\pi(s) = \arg\max_{a \in A} Q\left(s,a,\theta^{Q}\right)$$

3　仿真分析

为验证本文算法的有效性，将其与传统全部卸载、DQL-based（deep Q learning-based）算法[20]和SAC（soft actor-critor）算法[21]进行仿真对比。考虑2个环境场景下的地面用户：密集区域用户和稀疏区域用户，且2个区域的用户之间相互独立、互不干扰。仿真中所有地面基站均可覆盖所有密集区域用户，所有卫星均可覆盖所有用户。地面基站初始时均匀地分布在环境中，且位置固定不变。地面用户初始时随机散落在环境中，考虑用户为动态的情况，每个迭代间隙随机运动。用户携带的计算任务数据量大小为一个区间内的随机值，完成该任务所需的CPU周期总数也在一个区间内随机取样。

用户设定卸载任务的数据传输功率p以及等待MEC完成任务的待机功率p_t。具体的环境参数设计参考了文献[21-22]，如表1所示。

表1 可拆分任务卸载环境参数设计

参数	取值
地面基站数量N	5
LEO卫星数量M	3
LEO卫星轨道高度S_m/km	1 000
计算任务的数据大小D_i/KB	[800,1 200]
完成D_i所需的CPU周期总数X_i/MHz	[1 800,2 200]
工作频率f/GHz	1.2
传输功率p/mW	250
待机功率p_t/mW	50
MEC的计算能力$\frac{C_m}{C_n}$/GHz	20
UE计算能力C_i/GHz	2
地面信道带宽B_n/MHz	2
卫星信道带宽B_m/MHz	5
信道中的高斯白噪声/dBm	−100
路径损耗a_i	[2,4]

通过用户平均服务时延、因卸载而受益的地面用户数量以及地面用户任务卸载总开销等角度来评判本文算法的有效性，并选取以下几种卸载方案进行对比。

1) Full-Local。Full-Local表示每个用户的计算任务完全由用户本地计算资源计算处理，不再进行任务卸载工作，将其作为基准方案，用于评估星地融合网络中的任务卸载是否能够显著提升用户性能。

2) DQL-based。文献[18]中的深度Q学习（DQL）算法适用于离散任务卸载场景，特别是可分解任务的卸载优化。DQL-based算法具有较强的适应性，能够在星地融合网络中高效地处理任务卸载决策，特别适用于离散空间下的任务卸载和资源分配问题。

3) SAC。SAC算法能够处理连续资源分配问题，尤其适用于动态、连续的任务卸载场景。星地融合网络中的动态环境和多样化服务需求要求高效的连续任务卸载策略。SAC算法通过最大化期望回报和熵平衡，实现了在动态星地融合网络场景中的有效资源调度，具体的参数设置参考文献[19]。

STO-DDPG算法的相关参数如表2所示。

表2 STO-DDPG算法相关参数

参数	取值
折扣因子γ	0.99
软更新学习率τ	0.005
经验池Buffer容量	10 000
batch大小	100
随机噪声方差σ	1
方差递减值σ_t	0.999 9

1) 算法收敛性

在算法收敛性方面，图3分析了STO-DDPG算法累计总奖励回报值收敛情况，图4分析了STO-DDPG算法系统总服务时延收敛情况，用户数量为50个。从图3中可以看出，累计总奖励回报值随着训练周期的增加不断上升，并在200个周期之后趋于稳定。从图3中也可以看出，STO-DDPG算法较SAC和DQL-based算法收敛较快，且最终收敛奖励最大。从图4还可以看出，在大约200个周期之后，系统总服务时延稳定在最小值附近。在前100个周期内累计总奖励回报值和总服务时延的波动较大，这是因为在训练初期模型还不稳定，此时的干扰噪声对模型的训练影响较大。为了避免局部最优，此时算法更多地趋向于对环境的探索，随着学习不断进行，算法也会不断朝着探索最大的奖励回报值方向收敛。因此，综合图3和图4可以看出，STO-DDPG算法有更好的收敛性和稳定性。

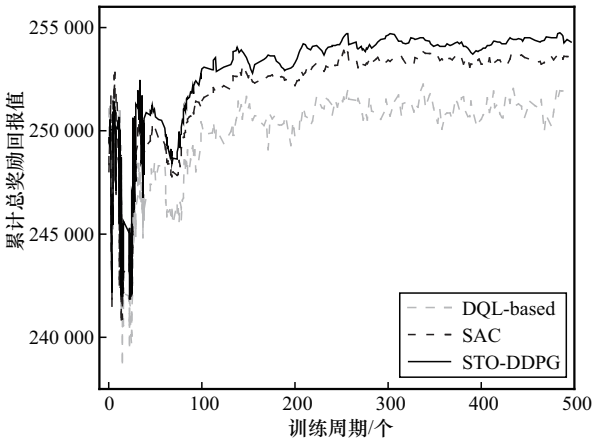

图3 STO-DDPG算法累计总奖励回报值收敛情况

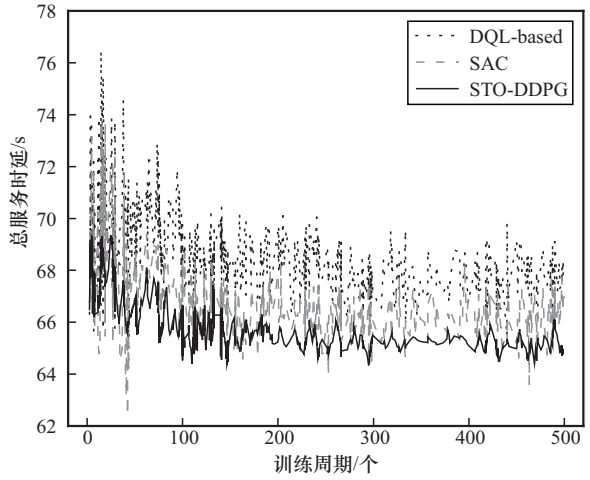

图4　STO-DDPG算法系统总服务时延收敛情况

2) 不同用户数量下用户平均服务时延

图5对比了几种卸载算法在用户数量不断增加时用户平均服务时延的变化情况。因为Full-Local算法的用户平均服务时延与用户数量无关，仅与每个用户自身的任务量大小和本地计算能力相关，因此将Full-Local算法的用户平均服务时延作为基准线。随着用户数量的增加，用户平均服务时延均呈现上升趋势。在用户数量较少时，STO-DDPG、DQL-based以及SAC算法的用户平均服务时延差异不大，这是因为在用户数量较少时，计算资源还较为充足，用户产生的计算任务可以得到及时且高效地处理。在用户数量增加至50个之后，STO-DDPG算法与其他2种算法的用户平均服务时延差距明显。

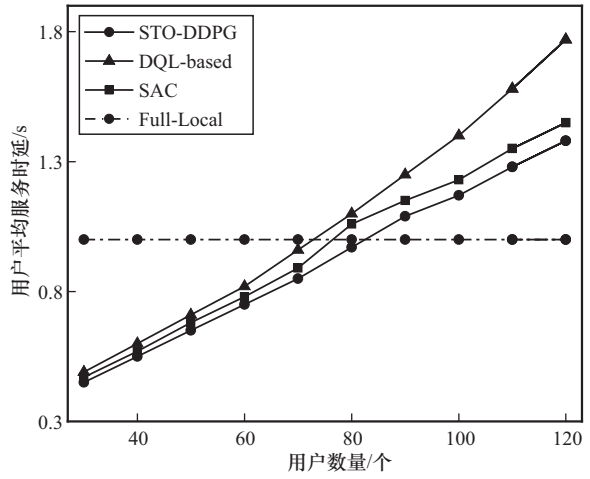

图5　用户数量与用户平均服务时延的关系

具体来说，在用户数量为70时，与SAC算法相比，STO-DDPG算法时延降低约7.05%；与DQL-based算法相比，STO-DDPG算法时延降低约12.6%。这是因为STO-DDPG算法可以感知全局环境状态，通过分析其他智能体的潜在策略来优化每个智能体的行动策略，因此STO-DDPG算法能更好适应较复杂的环境。

同时应该注意到，当用户数量增加到90个之后，3种算法的用户平均服务时延都要高于Full-Local算法，这主要是因为卫星网络的加入，星地之间的长传输时延是无法避免的，随着用户数量增加，不仅稀疏区域用户选择卫星MEC卸载任务，越来越多的密集区域用户也会因为计算资源的不足选择卫星MEC服务器，因此系统总服务时延会显著增加。显然，卫星网络在分担地面压力的同时能以更长的时延换取更大面积的覆盖能力。

3) 不同MEC服务器计算能力下用户平均服务时延

图6展示了上述几种卸载算法在不同MEC服务器计算资源时的用户平均服务时延变化情况，此时用户数量设置为50个。通过对比可以看出，除了Full-Local算法的用户平均服务时延不受MEC计算资源的影响之外，其他3种算法的用户平均服务时延均随着MEC计算资源的增强而不断下降。

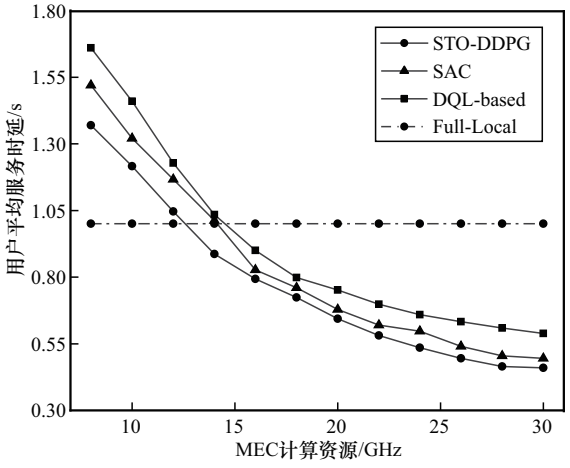

图6　MEC计算资源与用户平均服务时延的关系

在MEC计算资源较少时，3种算法的用户平均服务时延都高于Full-Local算法，这是因为MEC计算资源过少，不仅无法及时处理用户产生的计算任务，同时还增加了传输时延。随着MEC计算资源增加，用户平均服务时延也在不断降低。在MEC计算资源为30 GHz时，STO-DDPG算法

相较于SAC算法，用户平均服务时延降低7.4%，相较于DQL-based算法，用户平均服务时延降低29.6%。这是因为当MEC计算资源充足时，STO-DDPG算法可以感知全局信息优化卸载策略，同时可以将任务拆分卸载到卫星MEC进行处理，充分利用了增加的MEC计算资源，有效地提高了计算效率。从图6中也可以看出，随着MEC计算资源的增加，STO-DDPG算法对比SAC算法和DQL-based算法的优势更明显。

4) 不同用户数量下受益的用户数量

图7对比了不同算法随着用户数量增加，受益用户数量的变化情况。在这里，对“受益用户”的定义是通过执行任务卸载之后，总服务时延小于完全本地处理时延的用户。

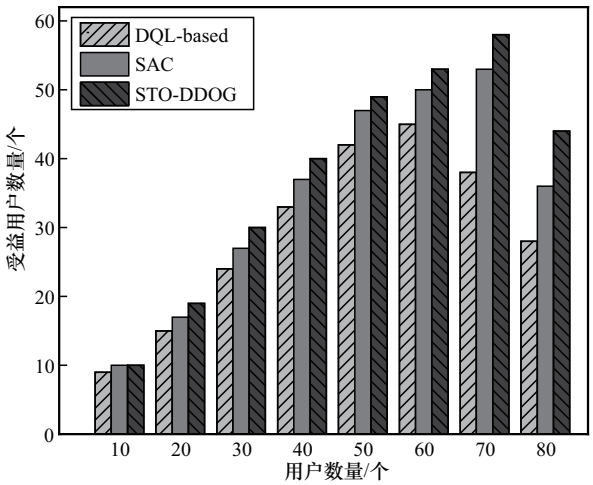

图7　用户数量与受益用户数量的关系

图5与图6中的用户平均服务时延是相对于整个系统的指标，而受益用户数量则表示的是每个用户个体的收益情况，是个体的指标。结果表明，SAC算法在用户数量大于70个之后受益用户数量就会显著减少。DQL-based算法在用户数量60个之后也会快速地降低。而STO-DDPG算法的受益用户数量指标虽然在用户数量达到临界值之后也会下降，不过该算法的受益用户数量会明显高于其他2种算法。这是因为随着用户数量的增加，用户生成的计算任务也会增多，STO-DDPG算法能感知全局信息并通过任务拆分，能够更有效地利用MEC计算资源。这种策略使得STO-DDPG算法在处理大量用户和计算任务时，能够保持较高的受益用户数量，从而表现出其优越的性能。

5) 不同用户数量下用户总开销

图8是随着用户数量增加，系统用户总开销的变化情况。从图8中可以看出，3种算法在用户数量增加时，整个系统的用户总开销也随之增加，并且在每组用户数量下，STO-DDPG算法用户总开销均小于SAC算法和DQL-based算法。由式(19)可知，用户总开销由本地计算开销、子任务传输开销和子任务处理等待开销3个部分组成。STO-DDPG算法通过将复杂大型的计算任务拆分为多个子任务并卸载到本地或卫星MEC并行处理，同时利用全局信息优化卸载策略，减少了子任务传输和子任务处理等待的开销，有效降低了用户总开销成本。

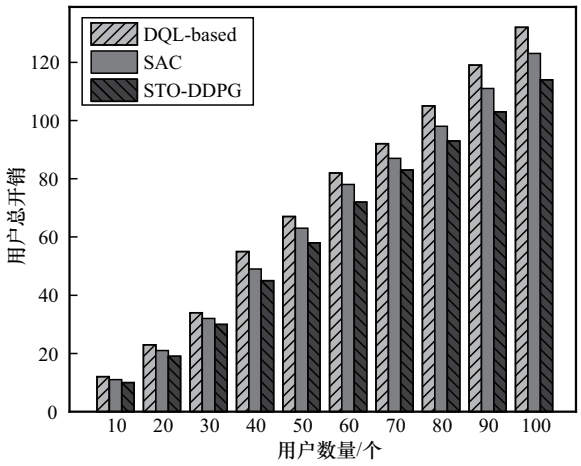

图8　用户数量与用户总开销的关系

综合上述图3~图8的仿真结果可以看出，本文算法STO-DDPG通过与多种任务卸载算法对比，不仅表现出很好的收敛性和稳定性，且在针对整个系统的用户平均服务时延及每个用户的个体受益情况中，均优于其他几种算法。同样在用户总开销方面，STO-DDPG算法表现得更好，通过优化任务分配和资源利用，STO-DDPG算法不仅提高了计算任务的处理效率，还优化了资源消耗，在提升服务质量的同时减轻了网络负担。

4　结束语

针对星地融合网络中的任务卸载时延长的问题，本文提出了一种基于深度确定性策略梯度的星地融合网络可拆分任务卸载算法STO-DDPG。首先，针对星地融合网络中地面网络密集和地面网络稀疏两类用户，建立了星地融合网络多接入边缘计算结构模型，并给出了基于不同任务卸载节点的服

务时延计算表达式。然后，将系统所有用户的总服务时延最小值定义为优化目标函数，基于DDPG算法设计了用户策略网络作为智能体，集合所有用户子任务分配策略为动作，计算资源分配矩阵为状态，并设计奖励函数以优化训练DDPG网络参数。通过与环境的持续交互，最终实现了优化问题的最优解。最后，设计了多组仿真对比实验，不仅分析了STO-DDPG算法的收敛性能，还从收敛性、稳定性以及能效性多个角度对比了其与Full-Local算法、SAC算法及基于DQL的DQL-based算法的性能差异。仿真结果表明，STO-DDPG算法在各项性能指标上均表现优越，验证了其有效性。

参考文献：

[1] 杨力, 潘成胜, 孔相广, 等. 5G融合卫星网络研究综述[J]. 通信学报, 2022, 43(4): 202-215.
YANG L, PAN C S, KONG X G, et al. Review on 5G-satellite integrated network[J]. Journal on Communications, 2022, 43(4): 202-215.

[2] RANAWEERA P, JURCUT A, LIYANAGE M. MEC-enabled 5G use cases: a survey on security vulnerabilities and countermeasures[J]. ACM Computing Surveys, 2022, 54(9): 1-37.

[3] SPINELLI F, MANCUSO V. Toward enabled industrial verticals in 5G: a survey on MEC-based approaches to provisioning and flexibility[J]. IEEE Communications Surveys & Tutorials, 2021, 23(1): 596-630.

[4] CHIEN W C, LAI C F, HOSSAIN M S, et al. Heterogeneous space and terrestrial integrated networks for IoT: architecture and challenges[J]. IEEE Network, 2019, 33(1): 15-21.

[5] MOHAMMED A S, VENKATACHALAM K, HUBÁLOVSKÝ S, et al. Smart edge computing for 5G/6G satellite IOT for reducing inter transmission delay[J]. Mobile Networks and Applications, 2022, 27(3): 1050-1059.

[6] RODRIGUES T K, KATO N. Network slicing with centralized and distributed reinforcement learning for combined satellite/ground networks in a 6G environment[J]. IEEE Wireless Communications, 2022, 29(1): 104-110.

[7] 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.

[8] LIAO H J, ZHOU Z Y, ZHAO X W, et al. Learning-based queue-aware task offloading and resource allocation for space-air-ground-integrated power IoT[J]. IEEE Internet of Things Journal, 2021, 8(7): 5250-5263.

[9] SHI J F, YANG H S, CHEN X, et al. Resource allocation for integrated satellite-terrestrial networks based on RSMA[J]. IET Communications, 2024.

[10] TANG S Q, PAN Z S, HU G Y, et al. Deep reinforcement learning-based resource allocation for satellite Internet of things with diverse QoS guarantee[J]. Sensors, 2022, 22(8): 2979.

[11] CHAI F R, ZHANG Q, YAO H P, et al. Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT[J]. IEEE Transactions on Vehicular Technology, 2023, 72(6): 7783-7795.

[12] TANG Q Q, FEI Z S, LI B, et al. Computation offloading in LEO satellite networks with hybrid cloud and edge computing[J]. IEEE Internet of Things Journal, 2021, 8(11): 9164-9176.

[13] 黄韬, 刘江, 汪硕, 等. 未来网络技术与发展趋势综述[J]. 通信学报, 2021, 42(1): 130-150.
HUANG T, LIU J, WANG S, et al. Survey of the future network technology and trend[J]. Journal on Communications, 2021, 42(1): 130-150.

[14] CHEN D, WANG W, JIANG T. New multicarrier modulation for satellite-ground transmission in space information networks[J]. IEEE Network, 2020, 34(1): 101-107.

[15] KALLFASS I, HENNEBERGER R, SOMMER R, et al. High system gain E-band link in a wideband aircraft-to-ground data transmission[C]// Proceedings of the 2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS). Piscataway: IEEE Press, 2019: 1-5.

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

[17] 时圣苗, 刘全. 采用分类经验回放的深度确定性策略梯度方法[J]. 自动化学报, 2022, 48(7): 1816-1823.
SHI S M, LIU Q. Deep deterministic policy gradient with classified experience replay[J]. Acta Automatica Sinica, 2022, 48(7): 1816-1823.

[18] XING X Y, WANG S Y, LIU W J. An improved DDPG and its application in spacecraft fault knowledge graph[J]. Sensors, 2023, 23(3): 1223.

[19] QIU C, YAO H P, YU F R, et al. Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(6): 5871-5883.

[20] ZHANG R T, ZHAO B. Task offloading and resource allocation for cloud-edge collaboration in low earth orbit satellite networks[C]//Proceedings of the 2023 IEEE 23rd International Conference on Communication Technology (ICCT). Piscataway: IEEE Press, 2023: 764-769.

[21] LI Y L, LIANG L, FU J L, et al. Multiagent reinforcement learning for task offloading of space/aerial-assisted edge computing[J]. Security and Communication Networks, 2022, 2022: 4193365.

[22] AL-HOURANI A, GUVENC I. On modeling satellite-to-ground pathloss in urban environments[J]. IEEE Communications Letters, 2021, 25(3): 696-700.

[作者简介]

宋晓勤（1973– ），女，江苏扬州人，博士，南京航空航天大学副教授、硕士生导师，主要研究方向为智能组网与协同技术等。

吴志豪（2000– ），男，湖南邵阳人，南京航空航天大学硕士生，主要研究方向为多接入边缘计算、多目标资源联合分配等。

赖海光（1975– ），男，江苏南京人，南京控维通信科技有限公司高级工程师，主要研究方向为卫星通信技术等。

雷磊（1981– ），男，江西南昌人，博士，南京航空航天大学教授、博士生导师，主要研究方向为航空平台组网技术、智能无人机集群技术等。

张莉涓（1988– ），女，四川广安人，博士，南京航空航天大学副教授、硕士生导师，主要研究方向为无线泛在网络接入技术、智能无人机集群技术等。

吕丹阳（1999– ），男，山东青岛人，南京航空航天大学硕士生，主要研究方向为多目标强化学习训练、多目标资源联合分配等。

郑成辉（1977– ），男，江苏南京人，南京控维通信科技有限公司工程师，主要研究方向为卫星通信技术等。