基于生成对抗网络辅助多智能体强化学习的边缘计算网络联邦切片资源管理

林 艳 夏开元 张一晋*

(南京理工大学电子工程与光电技术学院 南京 210094)

摘 要：为满足动态边缘计算网络场景下用户差异化服务需求，该文提出一种基于生成对抗网络(GAN)辅助多智能体强化学习(RL)的联邦切片资源管理方案。首先，考虑未知时变信道和随机用户流量到达的场景，以同时优化长期平均服务等待时延和服务满意率为目标，构建联合带宽和计算切片资源管理优化问题，并进一步建模为分布式部分可观测马尔可夫决策过程 (Dec-POMDP)。其次，运用多智能体竞争双深度Q网络(D3QN)方法，结合GAN算法对状态值分布多模态学习的优势，以及利用联邦学习框架促使智能体合作学习，最终实现仅需共享各智能体生成网络加权参数即可完成切片资源管理协同决策。仿真结果表明，所提方案相较于基准方案能够在保护用户隐私的前提下，降低用户平均服务等待时延28%以上，且同时提升用户平均服务满意率8%以上。

关键词：边缘计算；网络切片；多智能体强化学习；联邦学习；生成对抗网络

中图分类号：TN929.5 **文献标识码**：A **文章编号**：1009-5896(2025)03-0666-12

DOI: 10.11999/JEIT240773

1 引言

为满足计算密集型和时延敏感型业务的新一代通信服务需求[1]，边缘计算技术通过在靠近用户的网络边缘侧部署算力资源，实现算力网络部署从云计算中心下沉至网络边缘端，进而解决传统云计算无法满足海量实时数据处理诉求的难题。考虑到不同业务对时延、带宽和算力要求不同，如何管理和分配有限的通信和算力资源成为边缘计算网络的关键问题之一。但由于业务需求的多样性[2]，传统资源管理架构难以保障差异化服务要求[3]。比如，自动驾驶类业务要求毫秒级时延，而智慧城市类业务需支持高质量的宽带连接。网络切片技术[4]作为一种按需组网架构[5]，通过将网络资源划分成多个独立的网络子切片，使得不同网络子切片所使用的资源之间相互隔离，进而实现通信和计算资源的灵活分配和定制化管理，故已成为边缘计算网络提供差异化边缘服务的重要使能技术[6,7]。

受限于边缘服务器有限的通信、计算资源，为满足多样化的业务需求，边缘计算网络切片资源管理研究变得尤为关键。目前，国内外学者已经对边缘计算网络切片资源管理策略展开了大量的研究。比如，文献[8]针对网络切片辅助边缘计算系统中切片定价和资源分配策略问题，提出了一种基于斯塔克尔贝格博弈的资源管理方案，以满足不同用户设备的服务质量需求。文献[9]针对网络切片辅助边缘计算系统的联合切片选择和带宽、计算资源管理问题，以最小化计算任务完成时间为优化目标，提出了一种具有有界近似比的卸载切片选择算法。文献[10]针对车联网边缘计算环境中如何克服网络切片通信、计算资源调度时面临的车辆移动性问题，提出了一种适用于车辆移动性的网络切片通信、计算资源管理方案，提升了用户体验并降低了切片迁移延迟。然而，上述工作无法应对边缘计算网络中无线信道条件的时变性和业务到达的随机性。

为此，研究者们提出一些基于深度强化学习(Deep Reinforcement Learning, DRL)的智能切片资源管理方案, 旨在提升系统资源利用效率[11]。比如，文献[12]提出了基于Q-学习的联合功率和带宽切片资源分配算法，实现用户服务体验的提升；文献[13]通过将用户请求分为基于资源和基于速率的请求，进一步结合异步优势演员评论家(Asynchronous Advantage Actor Critic, A3C)算法实现功率和频谱切片资源的联合优化。鉴于传统DRL存在训练不稳定性和收敛速度慢的问题，文献[14]结合生成对抗网络(Generative Adversarial Network, GAN)提升DRL算法对动态环境的适应能力，提出了一种基于GAN辅助DRL的带宽切片资源分配算法，同时保证服务质量和提升系统吞吐量。但上述切片资源管理方案未考虑边缘计算网络环境。文献[15]考虑边缘计算网络环境中信道条件和用户流量到达的动态性，运用优势演员评论家(Advantage Actor Critic, A2C)算法实现切片间通信、计算资源联合

收稿日期：2024-09-09；改回日期：2025-02-27；网络出版：2025-02-28
*通信作者： 张一晋 yijin.zhang@gmail.com
基金项目：国家自然科学基金(62001225, 62071236)
Foundation Items: The National Natural Science Foundation of China (62001225, 62071236)

管理。文献[16]研究了雾无线接入网环境下网络切片通信、计算、缓存资源联合管理问题，并利用双深度Q网络(Double Deep Q Network, DDQN)算法来自适应地学习动态无线信道条件下的最优切片资源分配策略。

然而，上述基于单智能体的中心化切片资源管理策略仍需要将大量观测信息上传至云计算中心，并依赖于其算力完成对未知边缘计算网络环境知识的建模和学习。鉴于边缘计算网络资源部署的分布式特点，基于多智能体强化学习的切片资源管理成为一种可行的解决途径。例如，文献[17]提出了一种基于多智能体Q-学习的网络切片带宽资源管理方案，其中各切片作为智能体，通过各自观测的本地环境知识实现对切片间带宽资源的按需分配，有效降低了延迟并提升了系统吞吐量。考虑到用户流量到达的随机性，除了基站需对切片资源管理以外，切片内还需再次分配资源给各用户。于是，文献[18]提出一种基于双层DRL的资源管理方案，其中上层基站采用深度Q网络(Deep Q-Network, DQN)算法实现中心化切片间通信资源管理，而下层各切片运用多智能体DQN算法实现分布式切片内资源调度。文献[19]针对车联网环境下的网络切片资源分配问题，综合考虑切片运行成本、重配置成本和用户服务质量约束，基站通过调整带宽、计算资源分配策略并运用多智能体深度确定性策略梯度算法，最终实现系统服务满意率的提升。文献[20]针对车联网边缘计算场景下基站网络切片通信、计算资源分配和负载平衡问题，利用双层约束强化学习算法，有效平衡基站的工作负载，并且提升用户平均满意率。

综上所述，边缘计算场景下的网络切片资源管理方案主要侧重于联合优化带宽和计算资源，以提高系统吞吐量以及资源利用效率。尽管现有多智能体切片资源管理方案解决了全局环境知识难以获取导致非最优策略选择的问题，但跨智能体间的合作学习仍需依赖于大量学习经验信息的共享，从而增加通信开销，且带来泄露用户隐私的风险。而联邦学习允许在保护用户隐私的前提下，使得多个智能体之间进行模型训练[21]，无需直接共享训练数据，因此多智能体DRL算法与联邦学习的结合成为保护用户隐私约束下资源管理问题的可行解决途径[22]。此外，上述基于值学习DRL的方案存在训练收敛速度慢、对高维状态空间的适应能力不佳以及易受局部最优解影响的问题。而值分布DRL算法通过将每个状态下的状态动作值看作一个分布，能够更好地捕捉到状态动作值的不确定性和分布情况，故有望缓解传统值学习算法对状态动作值的高估问题。

基于上述背景，本文针对边缘计算场景下的网络切片资源管理问题，研究如何通过联合优化切片间的带宽和计算资源管理策略，实现长期平均服务等待时延最小化和平均服务满意率最大化。与已有工作[14]相比，本文进一步考虑边缘计算场景，并运用多智能体深度强化学习方法，以实现各智能体切片资源管理的协同决策。此外，为解决传统多智能体强化学习在边缘计算动态环境下的难收敛和隐私泄露问题，本文一方面考虑利用GAN通过生成状态动作值分布以增强传统多智能体强化学习方法对状态值信息的学习能力；另一方面通过引入联邦学习架构，在保护用户隐私的同时实现智能体间学习经验共享，提升收敛性能。本文的创新工作主要包括：

(1) 针对未知时变信道条件和随机用户流量到达的边缘计算网络环境，以最小化长期平均服务等待时延并最大化平均服务满意率为优化目标, 建立了联合带宽、计算的网络切片资源管理决策问题，并进一步将该问题建模为分布式部分可观察马尔可夫决策过程(Decentralized Partially Observable Markov Decision Process, Dec-POMDP)。

(2) 提出一种基于GAN辅助多智能体强化学习的边缘计算网络联邦切片资源管理方案，通过在多智能体框架下运用竞争双深度Q网络(Dueling Double Deep Q-Network, D3QN)算法对状态动作价值进行拆分，利用联邦学习架构共享智能体学习经验，再结合GAN对状态值分布多模态学习的优势，实现在仅需共享各智能体生成网络加权参数即可完成切片资源管理协同决策。

(3) 仿真结果验证了所提方案能在保护用户隐私的同时，有效降低长期平均服务等待时延和提高平均服务满意率。此外，仿真结果还表明了在不同智能体数目和用户数目的情况下，所提方案的效能均明显优于基准方案。

2　系统模型

本节介绍网络切片辅助的边缘计算网络系统模型，包括网络模型、网络切片模型、无线通信模型和计算卸载模型。

2.1 网络模型

考虑如图1所示的云中心节点和B个接入点(Access Point, AP)的网络切片辅助边缘计算网络系统，其中每个AP均配置边缘服务器，且可以向其通信范围内的用户同时提供通信和计算服务。令AP集合为$\mathcal{B}=\{1,2,\cdots,B\}$，在基站$b$服务范围内均匀分布的用户集合为$\mathcal{U}_b=\{1,2,\cdots,U_b\}$。假设系统时间被

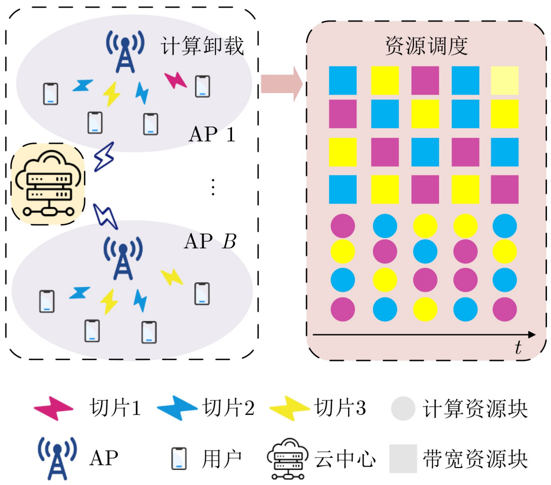

图 1 网络切片辅助的边缘计算网络系统模型

分成T个持续时间为τ的离散时隙，时隙集合为$\mathcal{T}=\{1,2,\cdots,T\}$。

2.2 网络切片模型

为向不同用户提供差异化服务质量，各AP在每个时隙开始时均需决策其关联切片间的通信和计算资源分配策略。假设共有I个网络切片，其集合为$\mathcal{I}=\{1,2,\cdots,I\}$。假设各AP的带宽资源$W_b$和计算资源$C_b$均被分为大小相同的若干份资源块。令每份带宽资源块的带宽大小为ρ^{B}，每份计算资源块包含的计算能力(即CPU频率)为ρ^{C}。令$f_{i,t,b}^{\mathrm{B}}$和$f_{i,t,b}^{\mathrm{C}}$分别表示AP b的网络切片i在时隙t的带宽和计算资源块数量，则$f_{i,t,b}^{\mathrm{B}}\in\left[1,W_b/\rho^{\mathrm{B}}\right]$，$f_{i,t,b}^{\mathrm{C}}\in\left[1,C_b/\rho^{\mathrm{C}}\right]$，且分别满足$f_{i,t,b}^{\mathrm{B}}\in N^+$和$f_{i,t,b}^{\mathrm{C}}\in N^+$。于是，$\boldsymbol{f}_{t,b}^{\mathrm{B}}=\left[f_{i,t,b}^{\mathrm{B}}\right]_{\forall i\in\mathcal{I}}$和$\boldsymbol{f}_{t,b}^{\mathrm{C}}=\left[f_{i,t,b}^{\mathrm{C}}\right]_{\forall i\in\mathcal{I}}$分别表示时隙$t$时AP b的各切片带宽和计算资源管理策略。若用户u_b在时隙t关联至切片i，则其分配的切片计算资源为

$$c_{u_b,t}=\frac{f_{i,t,b}^{\mathrm{C}}\rho^{\mathrm{C}}}{\sum\limits_{u_b\in\mathcal{U}_b}1_{\{k_{u_b,t==1}\}}} \tag{1}$$

其中，$k_{u_b,t}$为用户u_b在时隙t的切片选择索引号，$1_{\{\cdot\}}$表示在满足约束情况下置1，反之置0。

考虑到实际的网络环境中，由于带宽资源受到物理硬件的限制，难以实现完全精细化的平均分配。为此，本文进一步假设切片带宽资源离散化。假设各用户已按概率随机关联某一种切片服务，且选择相同切片的用户均分其带宽和计算资源。若用户u_b在时隙t关联至切片i，则其分配的切片带宽资源为

$$w_{u_b,t}=\bar{W}\left\lfloor\frac{f_{i,t,b}^{\mathrm{B}}\rho^{\mathrm{B}}}{\bar{W}\sum\limits_{u_b\in\mathcal{U}_b}1_{\{k_{u_b,t}==i\}}}\right\rfloor \tag{2}$$

其中，$\bar{W}$为带宽资源单元大小，$\lfloor\cdot\rfloor$为向下取整符号。

2.3 无线通信模型

假设各AP和用户均只配备单天线，且不同用户的链路已分配不同频谱资源块，故不同用户间干扰已消除。考虑信道功率增益由快衰落和慢衰落两部分组成，其中快衰落部分为瑞利衰落，慢衰落部分由路径损耗组成。假设各时隙内的信道状态保持不变，故时隙t时用户u_b和其服务AP b的上行信道增益$G_{u_b,t}^{\mathrm{up}}$可以表示为

$$G_{u_b,t}^{\mathrm{up}}=\left|h_{u_b,t}\right|^2g_{u_b,t} \tag{3}$$

其中，$h_{u_b,t}$为时隙t时用户u_b与AP b之间的瑞利信道系数，$g_{u_b,t}$为用户u_b与AP b之间链路在时隙t时的路径损耗。此时，用户u_b与AP b之间的上行传输速率为

$$r_{u_b,t}^{\mathrm{up}}=w_{u_b,t}\log_2\left(1+\frac{G_{u_b,t}^{\mathrm{up}}P^{\mathrm{U}}}{\delta^2}\right) \tag{4}$$

其中，P^{U}为各用户的传输功率，δ^2表示加性高斯白噪声的方差。

类似地，时隙t时AP b与用户u_b之间的下行传输速率为

$$r_{u_b,t}^{\mathrm{down}}=w_{u_b,t}\log_2\left(1+\frac{G_{u_b,t}^{\mathrm{down}}P^{\mathrm{A}}}{\delta^2}\right) \tag{5}$$

其中，$G_{u_b,t}^{\mathrm{down}}$为时隙$t$时用户$u_b$和AP b的下行信道增益，P^{A}为AP的传输功率。

2.4 计算卸载模型

假设各用户在每个时隙开始时刻均到达计算任务，且各计算任务均可在单个时隙持续时间内完成计算卸载。选择AP b的切片i的用户u_b在时隙t到达的计算任务表征为$\left\{s_{u_b,t},x_{u_b,t},y_{u_b,t},l_{k_{u_b,t}}^{\max}\right\}$，其中$s_{u_b,t}$为完成该计算任务所需要的CPU周期数，$x_{u_b,t}$为任务处理前的数据包大小，$y_{u_b,t}$为该任务处理后的数据包大小，$l_{k_{u_b,t}}^{\max}$为该用户的最大服务时延容忍阈值。特别地，$y_{u_b,t}=\beta x_{u_b,t}$，其中$\beta$为处理后数据包大小相较于处理前的比例。由于用户的本地计算能力有限，各用户需要将其到达的任务卸载至最近的AP进行计算。一般来说，计算卸载过程包括3个阶段：

(1)上传：当用户u_b的任务在时隙t到达时，先将数据包上传至AP b并由其内部署的边缘服务器处理计算任务，于是任务上传时延为$l_{u_b,t}^{\mathrm{up}}=x_{u_b,t}/r_{u_b,t}^{\mathrm{up}}$。

(2)边缘计算：当AP b接收到用户u_b的计算任务后，将调用该用户所选切片的计算资源进行处理，则此时用户u_b任务的计算时延为$l_{u_b,t}^{\mathrm{cal}}=s_{u_b,t}/c_{u_b,t}$。

(3)结果反馈：当计算任务完成后，AP b调用用户所选切片的带宽资源并将计算结果反馈至用户u_b，则此时结果反馈时延为$l_{u_b,t}^{\rm down}=y_{u_b,t}/r_{u_b,t}^{\rm down}$。

于是，用户u_b在时隙t时计算卸载的总时延为$l_{u_b,t}^{\rm total}=l_{u_b,t}^{\rm up}+l_{u_b,t}^{\rm cal}+l_{u_b,t}^{\rm down}$。需要注意的是，对于选择不同网络切片的各用户，其服务时延容忍度不同。令$l_i^{\max}\,(i\in\mathcal{I})$表示用户对切片$i$服务时延的最大容忍阈值，且令二进制变量$\mu_{u_b,t}$表征用户$u_b$在时隙$t$是否满足所选切片的服务时延要求，即若$\sum_i l_{u_b,t}^{\rm total}1_{\{k_{u_b,t}==i\}}\le l_{k_{u_b,t}}^{\max}$时，$\mu_{u_b,t}=1$；否则，$\mu_{u_b,t}=0$。

3　问题表述

3.1　优化目标

本文旨在通过联合设计各AP的带宽和计算切片资源管理策略，以同时实现用户长期平均服务等待时延的最小化和平均服务满意率的最大化。为此，首先分别定义平均等待时延和平均服务满意率性能指标。

(1) 平均服务等待时延：在网络切片辅助的边缘计算网络系统中，平均服务等待时延与AP的带宽、计算切片资源管理策略紧密相关[13]。故时隙t用户的平均服务等待时延为

$$L_t^{\rm Ave}=\frac{1}{B}\sum_{b=1}^{B}\left[\frac{1}{U_b}\sum_{u_b=1}^{U_b}l_{u_b,t}^{\rm total}\right]\tag{6}$$

(2) 平均服务满意率：考虑到不同切片对时延的要求不同，定义时隙t时用户的平均服务满意率为满足所选切片服务时延要求的用户数与总用户数之比，表示为

$$\varphi_t^{\rm Ave}=\frac{1}{B}\sum_{b=1}^{B}\left[\frac{1}{U_b}\sum_{u_b=1}^{U_b}\mu_{u_b,t}\right]\tag{7}$$

考虑到时变无线信道条件和随机用户流量到达的环境动态性，本文通过为各AP在每个时隙寻找最优的带宽、计算切片资源管理策略，即$\boldsymbol{f}_t^{\rm B}=\left[\boldsymbol{f}_{t,b}^{\rm B}\right]_{b\in\mathcal{B}}$和$\boldsymbol{f}_t^{\rm C}=\left[\boldsymbol{f}_{t,b}^{\rm C}\right]_{b\in\mathcal{B}}$，以权衡用户的平均服务等待时延和平均服务满意率的长期性能。因此，本文的优化问题可表示为

$$\min_{\boldsymbol{f}_t^{\rm B},\boldsymbol{f}_t^{\rm C}}\mathbb{E}\left[\alpha L_t^{\rm Ave}-(1-\alpha)\,\varphi_t^{\rm Ave}\right]\tag{8a}$$

$$\text{s.t.}\sum_{i=1}^{I}f_{i,t,b}^{\rm B}\rho^{\rm B}\le W_b,\forall t\in\mathcal{T},\forall b\in\mathcal{B}\tag{8b}$$

$$\sum_{i=1}^{I}f_{i,t,b}^{\rm C}\rho^{\rm C}\le C_b,\forall t\in\mathcal{T},\forall b\in\mathcal{B}\tag{8c}$$

其中，$\alpha\in[0,1]$为用户平均服务等待时延的权重系数，式(8b)和式(8c)分别表示所有已分配的带宽、计算切片资源总量不超过各AP所拥有的相应资源总量。

3.2　Dec-POMDP建模

在实际边缘计算网络环境中，受限于局部观测能力，单个AP智能体很难获得完整的状态信息。故可以采用多智能体强化学习方法来解决，其中每个AP智能体根据本地观测信息执行相应的本地动作决策，然后获得环境的奖励反馈。通过持续学习和优化动作决策，智能体能够获得更高的回报。为此，本文将优化问题建模为Dec-POMDP，表示为$\mathcal{B},\mathcal{S},\mathcal{O},\mathcal{A},\gamma,r$。其中，$\mathcal{B}$为AP智能体集合；$\mathcal{S}$表示有限状态空间集合；$\mathcal{O}$表示联合观测空间集合；$\mathcal{A}$表示联合动作空间集合；$\gamma$表示折扣因子；$r$表示系统奖励。其中，各智能体的局部观测、动作和奖励函数定义如下：

(1)局部观测：定义AP b在时隙t的观测包括其各切片待传输任务数据总量和待计算任务所需CPU周期数，分别表示为$\boldsymbol{M}_{b,t}=[m_{b,t,1},m_{b,t,2},\cdots,m_{b,t,I}]$和$\boldsymbol{N}_{b,t}=[n_{b,t,1},n_{b,t,2},\cdots,n_{b,t,I}]$。其中，$m_{b,t,i}=\sum_{u_b\in\mathcal{U}_b}(x_{u_b,t}+y_{u_b,t})\,1_{\{k_{u_b,t}==i\}}$表示AP b的切片i在时隙t的待传输任务数据总量，$n_{b,t,i}=\sum_{u_b\in\mathcal{U}_b}s_{u_b,t}1_{\{k_{u_b,t}==i\}}$表示AP b的切片i在时隙t待计算任务所需CPU周期数。于是，AP b在时隙t的观测可以表示为$\boldsymbol{o}_{b,t}=[\boldsymbol{M}_{b,t},\boldsymbol{N}_{b,t}]$。

(2)动作：在时隙t，AP b智能体根据获得的本地观测信息调整切片资源分配动作$\boldsymbol{a}_{b,t}=\left[\boldsymbol{f}_{b,t}^{\rm B},\boldsymbol{f}_{b,t}^{\rm C}\right]$，即包括AP b在时隙t的带宽和计算资源管理策略。

(3)奖励：为权衡用户的平均服务等待时延和平均服务满意率，时隙t的系统奖励由平均服务等待时延和平均服务满意率的效用函数组成。考虑到二者数值范围差异，本文将系统奖励设计为上述两种性能的指数效用函数加权和，表示为$r_t=\alpha\exp\left(-L_t^{\rm Ave}\right)+(1-\alpha)\exp\left(\varphi_t^{\rm Ave}\right)$。

4　基于GAN辅助多智能体强化学习的边缘计算网络联邦切片资源管理方案

传统多智能体强化学习算法在处理高维离散状态和动作空间时存在局限性，且跨智能体间的合作学习仍需依赖于大量学习经验信息的共享从而带来泄露用户隐私的风险。为此，本节首先运用多智能体强化学习框架在各智能体本地维护算法网络并与环境交互获取学习经验；然后结合D3QN算法对状态动作价值进行拆分，同时引入GAN方法的值分布学习机制[13]以增强D3QN算法在复杂边缘计算网络环境中的泛化能力，使其能够更好地适应不断变

化的信道条件和用户流量；最后引入联邦学习框架，允许各智能体在其本地数据上独立训练，仅通过聚合更新的方式共享全局信息，以减少通信开销和降低数据隐私泄露的风险。以下将先介绍D3QN-GAN算法基本原理，再详细阐述所提基于GAN辅助多智能体强化学习的联邦切片资源管理方案。

4.1 D3QN-GAN算法

D3QN算法作为DQN算法的扩展，通过引入结构相同的双神经网络(在线网络、目标网络)来解决DQN存在估计偏差和过高估计的问题。具体来说，在线网络用于选择动作$\boldsymbol{a}$和计算当前状态$\boldsymbol{s}$的状态动作值$Q(\boldsymbol{s},\boldsymbol{a})$，而目标网络则用于计算目标状态动作值$\hat{Q}(\boldsymbol{s},\boldsymbol{a})$，从而减少训练过程中Q值的波动性，提高训练的稳定性。此外，在线网络和目标网络的参数在训练过程中分开更新。

为了提高对Q值的估计效率和准确性，D3QN算法将Q值分解为状态值$V(\boldsymbol{s})$和动作优势值$A(\boldsymbol{s},\boldsymbol{a})$，即

$$Q(\boldsymbol{s},\boldsymbol{a})=V(\boldsymbol{s})+A(\boldsymbol{s},\boldsymbol{a})-\frac{1}{A}\sum_{\boldsymbol{a}'\in\mathcal{A}}A(\boldsymbol{s},\boldsymbol{a}')\tag{9}$$

其中，$V(\boldsymbol{s})$用于估计当前状态$\boldsymbol{s}$的价值；$A(\boldsymbol{s},\boldsymbol{a})$用于估计每个动作相对于其他动作的优势；最后一项则是通过减去动作优势函数均值解决D3QN网络收敛不唯一性的问题。

通过对Q值的分解，D3QN能够有效学习状态值和动作优势，从而帮助智能体更好地理解每个动作的效果和相对优势，提升在边缘计算网络环境中的学习效率和稳定性。在D3QN算法基础上，D3QN-GAN算法通过引入GAN算法对状态价值函数分布的学习，由生成网络输出状态值函数和动作优势函数，组合成真实的状态动作价值函数分布并交由判别网络进行对抗训练，从而改进模型的训练效果和决策策略。

具体来说，D3QN-GAN算法由一个主生成网络G和其对应的目标生成网络$\hat{G}$，以及一个判别网络组成D。生成网络G和$\hat{G}$均分为3个部分：前端的嵌入层、中间的多层全连接层和最后的分离层。嵌入层由负责接收状态$\boldsymbol{s}$的状态嵌入层和采样噪声$\boldsymbol{\tau}$的采样嵌入层组成；中间的全连接层用于增强算法的学习能力并传递信息；最后的分离层又可分为状态值分离层和优势值分离层两个模块。状态值分离层输出一组用于描述状态值分布的状态值粒子$\{G^{\mathrm{V}}(\boldsymbol{s},\boldsymbol{a})\}$，优势值分离层输出每个动作的优势值$G^{\mathrm{A}}(\boldsymbol{s},\boldsymbol{a})$。其中，生成网络生成的状态值粒子$\{G^{\mathrm{V}}(\boldsymbol{s},\boldsymbol{a})\}$由判别网络输出$D(\{G^{\mathrm{V}}(\boldsymbol{s},\boldsymbol{a})\})$。于是，智能体通过对生成网络输出进行处理获得Q值从而选择最优动作执行，得到环境奖励r并更新生成网络和判别网络。生成网络按照最小化判别网络输出值$D(\{G^{\mathrm{V}}(\boldsymbol{s},\boldsymbol{a})\})$方向进行梯度下降更新网络；目标生成网络$\hat{G}$每隔一段时间与主生成网络$G$同步；判别网络则以最大化真实状态值分布$r+\left\{\hat{G}^{\mathrm{V}}(\boldsymbol{s}',\boldsymbol{a}')\right\}$和生成状态值分布$\{G^{\mathrm{V}}(\boldsymbol{s},\boldsymbol{a})\}$的沃森斯坦距离为方向进行梯度上升更新网络。由此交替更新生成网络和判别网络，使生成网络能够生成更真实的Q值，从而收敛至最优。

4.2 基于多智能体联邦D3QN-GAN的边缘计算网络切片资源管理算法

为求解上述构建的 Dec-POMDP，本节将在上述D3QN-GAN算法基础上，结合多智能体DRL算法和联邦学习框架，提出一种基于多智能体联邦D3QN-GAN的边缘计算网络切片资源管理算法。如图2所示，每个AP智能体维护其各自的主生成网络G_b、目标生成网络$\hat{G}_b$和判别网络D_b，具体如下：

(1)生成网络：AP智能体b的生成网络G_b在时隙t接收观测$\boldsymbol{o}_{b,t}$和采样噪声$\boldsymbol{\tau}_{b,t}$后，输出N个状态值粒子$\left\{G_{b,t}^{\mathrm{V}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t})\right\}$和每个动作的优势值$G_{b,t,\boldsymbol{a}_{b,t}}^{\mathrm{A}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t}),\forall\boldsymbol{a}_{b,t}\in\mathcal{A}$。故相应状态动作价值函数$Q_{b,t}(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t})$可以表示为

$$\begin{aligned}Q_{b,t}(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t})=&\frac{1}{N}\sum G_{b,t}^{\mathrm{V}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t})\\&+G_{b,t,\boldsymbol{a}_{b,t}}^{\mathrm{A}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t})\end{aligned}\tag{10}$$

最大状态动作价值对应的动作，智能体可以选取最优动作执行。此外，目标生成网络$\hat{G}_b$接收状态信息$\boldsymbol{o}_{b,t}$后，输出N个目标状态值粒子$\left\{\hat{G}_{b,t}^{\mathrm{V}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t})\right\}$和每个动作的目标优势值$\hat{G}_{b,t,\boldsymbol{a}_{b,t}}^{\mathrm{A}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t})$。于是，目标状态动作价值函数$\hat{Q}_{b,t}(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t}),\forall\boldsymbol{a}_{b,t+1}\in\mathcal{A}$可表示为

$$\begin{aligned}\hat{Q}_{b,t}(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t})=&r_t+\frac{\gamma}{N}\sum\hat{G}_{b,t+1}^{\mathrm{V}}(\boldsymbol{o}_{b,t+1},\boldsymbol{\tau}_{b,t+1})\\&+\gamma\hat{G}_{b,t,\boldsymbol{a}_{b,t+1}}^{\mathrm{A}}(\boldsymbol{o}_{b,t+1},\boldsymbol{\tau}_{b,t+1})\end{aligned}\tag{11}$$

在训练过程中，各智能体会从本地经验回放池$\mathcal{M}_b=\{(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t},\boldsymbol{o}_{b,t+1},r_t)\}$中随机选取经验序列计算损失值$J_{b,t}^{G}$进行训练。生成网络$G_b$在时隙$t$的损失函数由判别网络$D_b$对其生成分布的判别项和贝尔曼均方误差项组成，即

$$\begin{aligned}J_{b,t}^{G}=&-\mathbb{E}_{(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t},\boldsymbol{o}_{b,t+1},r_t)\in\mathcal{M}_b}\\&\cdot\left[D_{b,t}\left(\left\{G_{b,t}^{\mathrm{V}}(\boldsymbol{o}_{b,t},\boldsymbol{\tau}_{b,t})\right\}\right)\right]-\frac{1}{2}\xi^2\end{aligned}\tag{12}$$

其中，$\xi=Q_{b,t}(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t})-\hat{Q}_{b,t}(\boldsymbol{o}_{b,t},\boldsymbol{a}_{b,t})$，$1/2\xi^2$为目标动作价值函数$\hat{Q}_{b,t}$和动作价值函数$Q_{b,t}$之间的贝尔曼均方误差。令生成网络$G_b$在时隙$t$的加权参数为$\theta_{b,t}^{G}$，则$\theta_{b,t}^{G}$的更新公式为

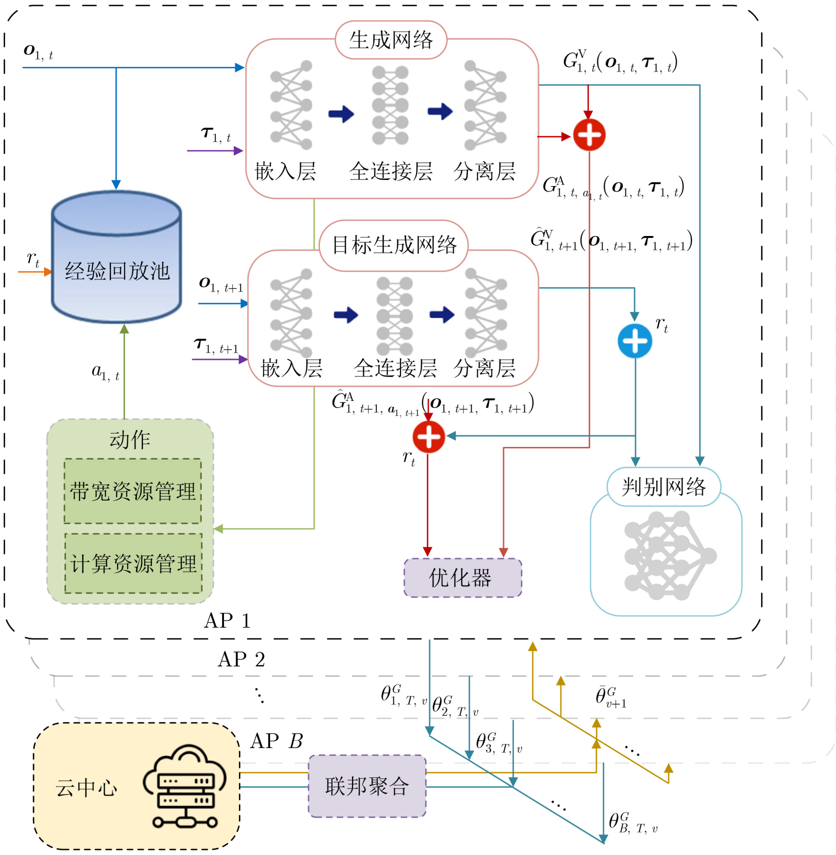

图 2 基于多智能体联邦D3QN-GAN的边缘计算网络切片资源管理算法原理图

$$\theta_{b,t+1}^{G} \leftarrow \theta_{b,t}^{G} - \eta^{G} \nabla_{\theta} J_{b,t}^{G} \tag{13}$$

其中，η^{G}为G_b网络的学习率，$\nabla_{\theta} J_{b,t}^{G}$为主生成网络$G_b$损失函数的梯度。目标生成网络$\hat{G}_b$并不主动参与训练，而是在主网络更新若干步后被动更新，可以表示为：$\hat{\theta}_{b,t}^{G} \leftarrow \theta_{b,t}^{G}$。

(2)判别网络：判别网络D_b由多层全连接层组成，当输入状态值分布值后可得到判别值。判别网络D_b的损失函数由分布判别项和梯度惩罚项组成，表示为

$$\begin{aligned} J_{b,t}^{D} = & \mathbb{E}_{(\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}, \boldsymbol{o}_{b,t+1}, r_t) \in \mathcal{M}_b} \\ & \cdot \left[D_{b,t}\left(\left\{ G_{b,t}^{\mathrm{V}}\left(\boldsymbol{o}_{b,t}, \boldsymbol{\tau}_{b,t} \right) \right\} \right) \right] \\ & - \mathbb{E}_{(\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}, \boldsymbol{o}_{b,t+1}, r_t) \in \mathcal{M}_b} \\ & \cdot \left[D_{b,t}\left(r_t + \left\{ \hat{G}_{b,t+1}^{\mathrm{V}}\left(\boldsymbol{o}_{b,t+1}, \boldsymbol{\tau}_{b,t+1} \right) \right\} \right) \right] \\ & + p_{b,t}(\lambda) \end{aligned} \tag{14}$$

其中，$D_{b,t}\left(\left\{G_{b,t}^{\mathrm{V}}\left(\boldsymbol{o}_{b,t}, \boldsymbol{\tau}_{b,t}\right)\right\}\right)$和$D_{b,t}\left(\left\{\hat{G}_{b,t+1}^{\mathrm{V}}(\boldsymbol{o}_{b,t+1}, \boldsymbol{\tau}_{b,t+1})\right\}\right)$分别为$D_b$网络对$G_b$和$\hat{G}_b$生成分布的判别值。通过最大化真实状态值分布与生成状态值分布判别之间的差值，可以提升D_b网络的判别能力。为稳定D_b网络训练，定义$p_{b,t}(\lambda)$为梯度惩罚项[23]，表示为

$$p_{b,t}(\lambda) = \frac{\lambda}{2}\left(\left\| \nabla_{\psi_{b,t}} D\left(\psi_{b,t} \right) \right\|_2 - 1 \right)^2 \tag{15}$$

其中，$\psi_{b,t} = \varepsilon_{b,t}\left(r_t + \gamma G_{b,t}^{\mathrm{V}}\left(\boldsymbol{o}_{b,t+1}, \boldsymbol{\tau}_{b,t+1} \right) \right) + (1 - \varepsilon_{b,t}) \cdot \hat{G}_{b,t}^{\mathrm{V}}\left(\boldsymbol{o}_{bt}, \boldsymbol{\tau}_{b,t} \right), \varepsilon_{b,t} \in (0,1)$。与$G_b$网络类似，在时隙$t$判别网络$D_b$的参数$\theta_{b,t}^{D}$更新方式可以表示为

$$\theta_{b,t+1}^{D} \leftarrow \theta_{b,t}^{D} - \eta^{D} \nabla_{\theta} J_{b,t}^{D} \tag{16}$$

其中，η^{D}为D_b网络的学习率，$\nabla_{\theta} J_{b,t}^{D}$为判别网络$D_b$损失函数的梯度。

(3) 联邦聚合：为充分利用跨智能体间的合作学习同时避免隐私泄露的风险，本文引入了联邦聚合方法来保护本地原始训练数据，允许智能体只共享训练网络加权参数。在本文提出的框架中，AP智能体上传其生成网络G_b的加权参数，从而降低隐私泄露的风险并提升收敛性能。

具体而言，在每个回合(包括T个时隙)，AP智能体b先各自运用D3QN-GAN算法完成本地训练。经过T步训练后，每个AP智能体的生成网络加权参数将上传至云中心进行参数共享。接下来，云中心对收集到的所有智能体的生成网络加权参数进行联邦聚合，获得下一回合训练使用的全局生成网络加权参数$\bar{\theta}_{v}^{G}$，表示为

$$\bar{\theta}_{v}^{G} = \frac{1}{B} \sum_{b=1}^{B} \theta_{b,T,v}^{G} \tag{17}$$

其中，$\bar{\theta}_v^G$为第v回合联邦聚合得到的全局生成网络加权参数，$\theta_{b,T,v}^G$为AP智能体b在回合v的第T步时的生成网络加权参数。之后，云中心将全局生成网络加权参数$\bar{\theta}_v^G$广播给所有AP智能体，作为下一回合的生成网络加权参数初始值，即

$$\theta_{b,1,v+1}^G = \bar{\theta}_v^G \tag{18}$$

4.3 训练过程

如算法1所示，所提算法训练过程如下：首先，初始化所有智能体的网络参数，包括生成网络、目标生成网络和判别网络并对回合索引T^{train}置0。在每个回合开始时，重置边缘计算网络环境。接着，在每个时隙内，AP智能体$b \in \mathcal{B}$会在获得本地观测$\boldsymbol{o}_{b,t}$和采样噪声$\boldsymbol{\tau}_{b,t}$后，计算得到状态动作价值函数$Q_{b,t}\left(\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}\right)$，并从中选出最优动作$\boldsymbol{a}_{b,t}^*$执行。之后，智能体得到环境奖励$r_t$和下一时隙的观测$\boldsymbol{o}_{b,t+1}$并存储训练信息$(\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}, \boldsymbol{o}_{b,t+1}, r_t)$至本地经验回放池$\mathcal{M}_b$。当回合数大于预设的训练回合数$T^{\text{update}}$后，网络开始训练。

在训练开始时，AP智能体$b \in \mathcal{B}$体根据从本地经验回放池$\mathcal{M}_b$随机抽取的小批经验，得到$D_{b,k}\left(\left\{G_{b,k}^{\mathrm{V}}\left(\boldsymbol{o}_{b,k}, \boldsymbol{\tau}_{b,k}\right)\right\}\right)$和$D_{b,k}\left(\left\{\hat{G}_{b,k}^{\mathrm{V}}\left(\boldsymbol{o}_{b,k}, \boldsymbol{\tau}_{b,k}\right)\right\}\right)$，再根据式(14)计算损失值$J_{b,k}^D$并更新判别网路$D_b$。之后，从$\mathcal{M}_b$中随机抽取小批经验，得到$D_{b,k}\left(\left\{G_{b,k}^{\mathrm{V}}\left(\boldsymbol{o}_{b,k}, \boldsymbol{\tau}_{b,k}\right)\right\}\right)$，并根据式(10)–式(11)计算状态动作价值函数$Q_{b,k}\left(\boldsymbol{o}_{b,k}, \boldsymbol{a}_{b,k}\right)$和目标状态动作价值函$\hat{Q}_{b,k}\left(\boldsymbol{o}_{b,k}, \boldsymbol{a}_{b,k}\right)$。然后，计算$G_b$网络损失值$J_{b,k}^G$并根据式(13)更新$G_b$网络。在$G_b$网络训练若干步后，复制$G_b$网络参数给$\hat{G}_b$网络。最后，在该回合的所有网络训练结束后，执行联邦聚合，并向各智能体广播下一回合生成网络参数$\bar{\theta}_v^G$。

算法 1 基于GAN辅助多智能体强化学习的边缘计算网络联邦切片资源管理算法

(1) 每个AP智能体初始化的生成网络G_b和判别网络D_b；

(2) 每个AP智能体初始化目标生成网络$\hat{G}_b$，本地经验回放池$\mathcal{M}_b$，粒子个数N；

(3) $T^{\text{train}} \leftarrow 0$；

(4) for Episode $v = 1, 2, \cdots, V$ do：

(5) 重置环境

(6) for TS $t = 1, 2, \cdots, T$ do：

(7) for AP智能体 $b = 1, 2, \cdots, B$ do：

(8) 采样噪声$\boldsymbol{\tau}_{b,t} \sim U(0,1)$，获取本地观测$\boldsymbol{o}_{b,t}$，同时输入生成网络$G_b$；

(9) 得到状态值粒子$\left\{G_{b,t}^{\mathrm{V}}\left(\boldsymbol{o}_{b,t}, \boldsymbol{\tau}_{b,t}\right)\right\}$和动作优势值$G_{b,t,\boldsymbol{a}_{b,t}}^{\mathrm{A}}\left(\boldsymbol{o}_{b,t}, \boldsymbol{\tau}_{b,t}\right)$

(10) 根据式(10)计算状态动作价值函数$Q_{b,t}\left(\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}\right)$；

(11) 执行动作$\boldsymbol{a}_{b,t}^* \leftarrow \operatorname{argmax} Q_{b,t}\left(\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}\right)$；

(12) 获取环境奖励r_t和下一时隙观测$\boldsymbol{o}_{b,t+1}$；

(13) 储存训练信息$\left\{\boldsymbol{o}_{b,t}, \boldsymbol{a}_{b,t}, \boldsymbol{o}_{b,t+1}, r_t\right\}$至本地经验回放池$\mathcal{M}_b$；

(14) end for；

(15) if $T^{\text{train}} \geq T^{\text{update}}$：

(16) for AP智能体 $b = 1, 2, \cdots, B$ do：

(17) 随机抽取 $\left\{\boldsymbol{o}_{b,k}, \boldsymbol{a}_{b,k}, \boldsymbol{o}_{b,k+1}, r_k\right\}_{k=1}^K \sim \mathcal{M}_b$，采样噪声$\left\{\boldsymbol{\tau}_{b,k}\right\}_{k=1}^K$和$\left\{\varepsilon_{b,k}\right\}_{k=1}^K$；

(18) 根据式(14)–式(16)计算损失函数$J_{b,k}^D$并根据式$\theta_{b,t+1}^D \leftarrow \theta_{b,t}^D - \eta^D \nabla_\theta J_{b,t}^D$ 更新网络D_b；

(19) 计算$Q_{b,k}\left(\boldsymbol{o}_{b,k}, \boldsymbol{a}_{b,k}\right)$和$\hat{Q}_{b,k}\left(\boldsymbol{o}_{b,k}, \boldsymbol{a}_{b,k}\right)$；

(20) 根据式(12)–式(13)计算损失函数$J_{b,k}^G$更新网络G_b，和网络$\hat{G}_b$；

(21) 根据式$\theta_{b,t+1}^G \leftarrow \theta_{b,t}^G - \eta^G \nabla_\theta J_{b,t}^G$更新主网络$G_b$，根据式$\hat{\theta}_{b,t}^G \leftarrow \theta_{b,t}^G \hat{\theta}_{b,t}^G \leftarrow \theta_{b,t}^G$更新目标网络$\hat{G}_b$；

(22) end for；

(23) end if；

(24) end for；

(25) 根据式(17)–式(18)执行联邦聚合并向所有智能体广播生成网络参数$\theta_{b,1,v+1}^G$；

(26) $T^{\text{train}} \leftarrow T^{\text{train}} + 1$；

(27) end for；

4.4 计算复杂度分析

考虑到每个智能体均包含主生成网络、目标生成网络和判别网络，而主、目标生成网络结构相同，因此本节分别分析生成网络和判别网络的计算复杂度。

(1)生成网络。假设各生成网络的嵌入层、隐藏层、状态值分离层和优势值分离层的神经元个数分别为D_1^G, D_2^G, D_3^G和D_4^G。假设生成网络输入状态维度为$D_{\boldsymbol{s}}^G$、噪声维度为D_n^G，输出状态值分布维度为D_{V}^G、输出动作优势值维度为D_{A}^G，则各生成网络计算复杂度为$\mathcal{O}\left(D_1^G D_{\boldsymbol{s}}^G + D_1^G D_n^G + D_1^G D_2^G + D_2^G D_3^G + D_3^G D_{\mathrm{V}}^G + D_2^G D_4^G + D_4^G D_{\mathrm{A}}^G\right)$。

(2)判别网络。假设判别网络为3层全连接层神经网络，其神经元个数分别为D_1^D, D_2^D, D_3^D。若判别网络输入维度为D_i^D，则其计算复杂度可以表示为$\mathcal{O}\left(D_i^D D_1^D + D_1^D D_2^D + D_2^D D_3^D + D_3^D\right)$。

5 仿真与分析

本节旨在通过数值仿真分析所提FedD3QN-GAN算法在网络切片辅助的边缘计算网络系统中的性能。在本文的仿真环境中，AP之间相距40 m呈线型排列，且每个AP 半径40 m覆盖范围内随机分布20名用户，用户位置服从均匀分布，用户切片选择概率分布为{0.46, 0.46, 0.08}。路径损耗模型采用

$g_{u_b,t}=128.1+37.6\lg d_{u_b,t}$，其中 $d_{u_b,t}$表示时隙t时用户u_b和 AP b之间的距离(km)。本文算法训练环境基于python仿真平台，其中每300回合网络学习率自适应调整为原来的0.1倍。其余默认仿真参数如表1所示。

为了验证本文所提方案的优势，本文采用3种基准方案进行对比：(1)基于多智能体独立D3QN-GAN算法的方案(ID3QN-GAN)：所有AP智能体独立训练本地D3QN-GAN网络，在每回合结束时仅更新本地网络参数。(2)基于多智能体联邦D3QN算法的方案(FedD3QN)：所有AP智能体独立训练本地D3QN网络，在每回合结束时上传D3QN网络参数并接收云中心聚合后的网络参数。(3)基于多智能体独立D3QN算法的方案(ID3QN)：所有AP智能体独立训练本地D3QN网络，在回合结束时仅更新本地网络参数。

图3对比了所有算法的平均累积奖励的收敛性。首先，随着训练回合数的增加，除ID3QN以外的所有算法的平均累积奖励不断增加直至收敛，分别验证了联邦多智能体框架和GAN算法在解决边缘计算网络切片资源管理问题的有效性。其次，本文算法和FedD3QN算法明显优于未采用联邦框架的两种算法。这表明引入联邦学习架构促使智能体分享彼此的学习经验，能够显著提升系统的收敛性能。此外，本文算法收敛后的平均累计奖励高于FedD3QN算法，这得益于GAN算法对值分布的学习使得智能体能够学习到多模态的状态动作值分布，从而增强智能体处理边缘计算网络环境中的多样性和复杂性的能力，进一步提升智能体的适应性和泛化能力。上述结果表明，本文算法能更好地适应动态信道条件和用户流量到达的边缘计算网络环境。

图4对比了在训练1 000回合后所有算法用户平均服务等待时延和平均服务满意率的性能权衡。首先，可以观察到本文算法在平均服务满意率和平均服务等待时延均明显优于其他算法，这是因为本文算法结合了联邦学习框架对动作空间的充分探索和GAN算法对值函数分布的充分学习的优势，从而实现高服务满意率与短服务等待时延的有效平衡。其次，ID3QN算法表现最差，这是因为该算法下独立训

表 1　仿真参数设置

系统参数	值
AP传输功率P^{A}	46 dBm
用户传输功率P^{U}	23 dBm
时隙持续时间τ	10 ms
最大可容忍时延$l_i^{\max}$	{5,8,9} ms
上行任务包大小$x_{u_b,t}$	{2.4,12,30} kbit
处理前后数据包之比β	0.25
计算任务量$s_{u_b,t}$	{0.1,0.2,1} kMc
用户数U_b	20
切片数量I	3
AP覆盖半径	40 m
AP数B	4
AP总带宽W_b	36 MHz
AP总计算资源C_b	900 kMc/s
带宽资源块大小ρ^{B}	2 MHz
计算资源块大小ρ^{C}	50 kMc
训练参数	值
生成网络学习率η^G	1e–3
判别网络学习率η^D	1e–3
奖励折扣系数 γ	0.8
每回合步数	100
状态值粒子个数N	30
权重系数α	0.5
批大小	32
经验回放池大小	50 000
目标网络更新频率	10τ
输入噪声维度	10

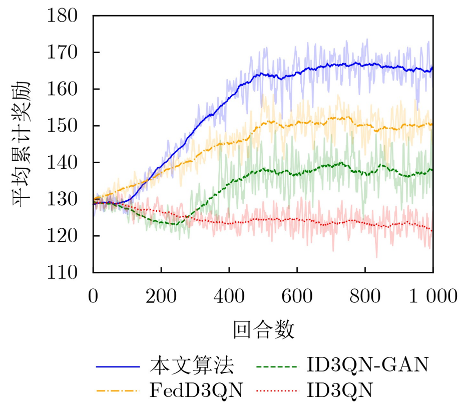

图 3 收敛性比较

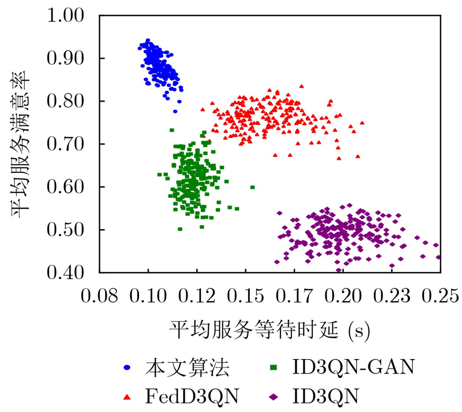

图 4 平均服务满意率与平均服务等待时延权衡

练的智能体无法共享学习经验，导致在处理复杂任务时缺乏足够的探索性和学习深度。此外，ID3QN-GAN算法在平均服务等待时延上优于FedD3QN算法，但在平均服务满意率上与FedD3QN算法差距较大。该现象表明FedD3QN算法倾向于以牺牲平均服务等待时间为代价换取更高的平均用户满意率，而ID3QN-GAN算法通过选择牺牲了部分用户的服务满意率实现更低的平均服务等待时延。

图5对比了在训练1 000回合后所有算法在不同用户数量下的用户平均服务满意率。首先可以观察到，随着用户数量的增加，所有算法的用户平均服务满意率均呈现下降趋势。这是由于在有限的带宽和计算资源下，用户数量的增加将不可避免地加剧资源竞争，造成部分用户获得的带宽和计算切片资源无法满足其服务时延的要求。其次，在不同用户数量下，本文算法满意率的均值均明显优于其他算法，且至少提升8%。该现象的原因一方面在于本文算法运用GAN算法直接从值函数分布中生成策略，故能学习到更全面的状态值信息，进而提升了决策策略的效率和准确性。另一方面，本文算法的AP智能体在每回合结束时通过共享网络参数可以将学到的经验迅速分享给其余智能体，从而进一步促进了智能体对动作空间的探索，加速全局收敛。以上现象表明，通过在多智能体DRL框架中引入联邦聚合机制并结合GAN算法的优势，可以使各智能体更高效地学习到最优切片资源管理策略。

图6对比了在训练1 000回合后不同AP数量对用户平均服务等待时延性能的影响。首先，从整体趋势观察可以发现，当AP智能体数由3增加到6时，本文算法的用户平均服务等待时延均显著低于其他

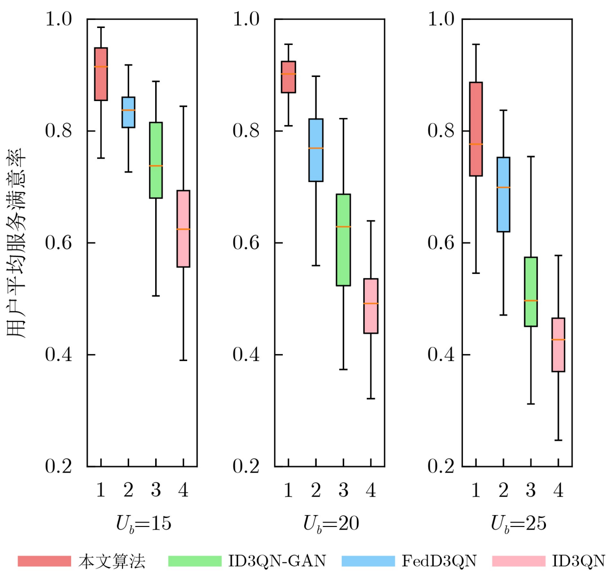

图 5 不同用户数量下用户平均满意率比较

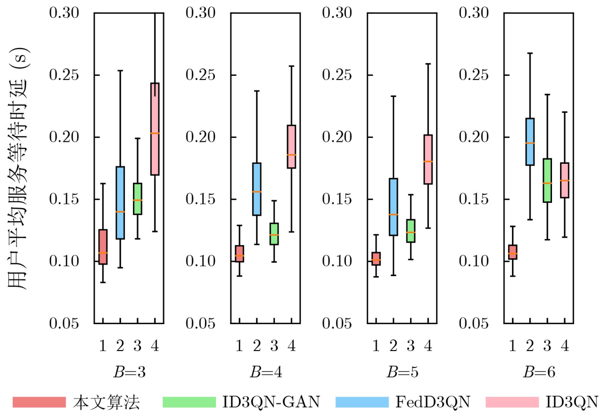

图 6 不同AP数量下用户平均服务等待时延比较

对比算法，且其平均值和方差也有明显下降(其均值相较于其余算法至少降低了28%)。该现象的原因在于，随着每回合结束时参与网络参数共享的AP智能体数量的增加，对动作空间和环境知识的探索也就更充分，有利于智能体选择最优动作执行以降低平均服务等待时延。其次，当AP智能体数由5增加到6时,非联邦的算法性能下降较为明显，而本文算法和FedD3QN算法的性能波动较小。其原因在于，随着参与网络参数共享的智能体数增加至一定数目后，单个智能体对于全局网络参数的贡献减小，这在某种程度上稀释了“优秀”智能体对系统奖励的贡献，从而造成了性能波动。综上，本文算法在不同智能体数量下均能保持显著的性能增益。

6 结束语

本文针对未知动态边缘计算网络场景中的联合带宽和计算切片资源管理问题，提出了一种基于GAN辅助多智能体强化学习的联邦切片资源管理方案。该方案能够增强智能体对状态值信息的学习能力，且通过共享智能体生成网络参数促使智能体合作学习，最终实现用户隐私约束下长期服务等待时延的降低和长期平均服务满意率的提升。仿真结果表明，所提方案的累计奖励收敛性能比基准方法提高了至少10%，且在不同用户数量下的平均服务满意度比基准方法高出8%以上。此外，所提方案在不同智能体数量下的平均服务等待时延比基准方法降低至少28%以上。然而，本研究仅考虑了理想的静态用户场景和无干扰通信条件。下一步工作将考虑更符合实际场景的动态性，如时变的用户移动性和复杂的多用户干扰等。

参 考 文 献

[1] GHONGE M, MANGRULKAR R S, JAWANDHIYA P M, *et al*. Future Trends in 5G and 6G: Challenges, Architecture, and Applications[M]. Boca Raton: CRC Press,

2022.

[2] DEBBABI F, JMAL R, FOURATI L C, *et al*. Algorithmics and modeling aspects of network slicing in 5G and Beyonds network: Survey[J]. *IEEE Access*, 2020, 8: 162748–162762. doi: 10.1109/ACCESS.2020.3022162.

[3] MATENCIO-ESCOLAR A, WANG Qi, and CALERO J M A. *SliceNetVSwitch*: Definition, design and implementation of 5G multi-tenant network slicing in software data paths[J]. *IEEE Transactions on Network and Service Management*, 2020, 17(4): 2212–2225. doi: 10.1109/TNSM.2020.3029653.

[4] 吴大鹏, 郑豪, 崔亚平. 面向服务的车辆网络切片协调智能体设计[J]. 电子与信息学报, 2020, 42(8): 1910–1917. doi: 10.11999/JEIT190635.
WU Dapeng, ZHENG Hao, and CUI Yaping. Service-oriented coordination agent design for network slicing in vehicular networks[J]. *Journal of Electronics & Information Technology*, 2020, 42(8): 1910–1917. doi: 10.11999/JEIT190635.

[5] 唐伦, 魏延南, 谭颀, 等. H-CRAN网络下联合拥塞控制和资源分配的网络切片动态资源调度策略[J]. 电子与信息学报, 2020, 42(5): 1244–1252. doi: 10.11999/JEIT190439.
TANG Lun, WEI Yannan, TAN Qi, *et al*. Joint congestion control and resource allocation dynamic scheduling strategy for network slices in heterogeneous cloud raido access network[J]. *Journal of Electronics & Information Technology*, 2020, 42(5): 1244–1252. doi: 10.11999/JEIT190439.

[6] SHAH S D A, GREGORY M A, and LI Shuo. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey[J]. *IEEE Access*, 2021, 9: 10903–10924. doi: 10.1109/ACCESS.2021.3050155.

[7] SHAH S D A, GREGORY M A, and LI Shuo. Toward network-slicing-enabled edge computing: A cloud-native approach for slice mobility[J]. *IEEE Internet of Things Journal*, 2024, 11(2): 2684–2700. doi: 10.1109/JIOT.2023.3292520.

[8] FAN Wenhao, LI Xuewei, TANG Bihua, *et al*. MEC network slicing: Stackelberg-game-based slice pricing and resource allocation with QoS guarantee[J]. *IEEE Transactions on Network and Service Management*, 2024, 21(4): 4494–4509. doi: 10.1109/TNSM.2024.3409277.

[9] JOŠILO S and DÁN G. Joint wireless and edge computing resource management with dynamic network slice selection[J]. *IEEE/ACM Transactions on Networking*, 2022, 30(4): 1865–1878. doi: 10.1109/TNET.2022.3156178.

[10] HUSAIN S, KUNZ A, PRASAD A, *et al*. Mobile edge computing with network resource slicing for internet-of-things[C]. The 2018 IEEE 4th World Forum on Internet of Things, Singapore, 2018: 1–6. doi: 10.1109/WF-IoT.2018.8355232.

[11] SHEN Xuemin, GAO Jie, WU Wen, *et al*. AI-assisted network-slicing based next-generation wireless networks[J]. *IEEE Open Journal of Vehicular Technology*, 2020, 1: 45–66. doi: 10.1109/OJVT.2020.2965100.

[12] ELSAYED M and EROL-KANTARCI M. Reinforcement learning-based joint power and resource allocation for URLLC in 5G[C]. 2019 IEEE Global Communications Conference, Waikoloa, USA, 2019: 1–6. doi: 10.1109/GLOBECOM38437.2019.9014032.

[13] AZIMI Y, YOUSEFI S, KALBKHANI H, *et al*. Energy-efficient deep reinforcement learning assisted resource allocation for 5G-RAN slicing[J]. *IEEE Transactions on Vehicular Technology*, 2022, 71(1): 856–871. doi: 10.1109/TVT.2021.3128513.

[14] HUA Yuxiu, LI Rongpeng, ZHAO Zhifeng, *et al*. GAN-powered deep distributional reinforcement learning for resource management in network slicing[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(2): 334–349. doi: 10.1109/JSAC.2019.2959185.

[15] ADDAD R A, DUTRA D L C, TALEB T, *et al*. Toward using reinforcement learning for trigger selection in network slice mobility[J]. *IEEE Journal on Selected Areas in Communications*, 2021, 39(7): 2241–2253. doi: 10.1109/JSAC.2021.3078501.

[16] LI Xuanheng, JIAO Kajia, CHEN Xingyun, *et al*. Demand-oriented Fog-RAN slicing with self-adaptation via deep reinforcement learning[J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(11): 14704–14716. doi: 10.1109/TVT.2023.3280242.

[17] ZHOU Hao, ELSAYED M, and EROL-KANTARCI M. RAN resource slicing in 5G using multi-agent correlated Q-learning[C]. The 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Finland, 2021: 1179–1184. doi: 10.1109/PIMRC50174.2021.9569358.

[18] AKYILDIZ H A, GEMICI Ö F, HÖKELEK I, *et al*. Hierarchical reinforcement learning based resource allocation for RAN slicing[J]. *IEEE Access*, 2024, 12: 75818–75831. doi: 10.1109/ACCESS.2024.3406949.

[19] CUI Yaping, SHI Hongji, WANG Ruyan, *et al*. Multi-agent reinforcement learning for slicing resource allocation in vehicular networks[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2024, 25(2): 2005–2016. doi: 10.1109/TITS.2023.3314929.

[20] HUANG Chen, CAO Jiannong, WANG Shihui, *et al*. Dynamic resource scheduling optimization with network coding for multi-user services in the internet of vehicles[J].

IEEE Access, 2020, 8: 126988–127003. doi: 10.1109/ACCESS.2020.3001140.

[21] LIN Yan, BAO Jinming, ZHANG Yijin, *et al*. Privacy-preserving joint edge association and power optimization for the internet of vehicles via federated multi-agent reinforcement learning[J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(6): 8256–8261. doi: 10.1109/TVT.2023.3240682.

[22] GUPTA A, MAURYA M K, DHERE K, *et al*. Privacy-preserving hybrid federated learning framework for mental healthcare applications: Clustered and quantum approaches[J]. *IEEE Access*, 2024, 12: 145054–145068. doi: 10.1109/ACCESS.2024.3464240.

[23] GULRAJANI I, AHMED F, ARJOVSKY M, *et al*. Improved training of Wasserstein GANs[C]. The 31st International Conference on Neural Information Processing Systems, Long Beach, USA, 2017: 5769–5779.

林 艳：女，副教授，研究方向为车联网、无人机通信等6G网络无线智能资源分配.

夏开元：男，硕士生，研究方向为网络切片资源分配与强化学习.

张一晋：男，教授，研究方向为序列设计、无线网络与人工智能.

责任编辑：余 蓉

Federated Slicing Resource Management in Edge Computing Networks based on GAN-assisted Multi-Agent Reinforcement Learning

LIN Yan XIA Kaiyuan ZHANG Yijin

(*School of Electronic and Optical Engineering*, *Nanjing University of Science and Technology*, *Nanjing* 210094, *China*)

Abstract:

Objective To meet the differentiated service requirements of users in dynamic Edge Computing (EC) network scenarios, network slicing technology has become a crucial enabling approach for EC networks to offer differentiated edge services. It facilitates flexible allocation and customized management of communication and computation resources by dividing network resources into multiple independent sub-slices. However, traditional slicing resource management methods cannot handle the time-varying wireless channel conditions and the randomness of service arrivals in EC networks. Additionally, existing intelligent slicing resource management schemes based on deep reinforcement learning face challenges, including the need for extensive information sharing, privacy leakage, and unstable training convergence. To address these challenges, the integration of Multi-Agent Reinforcement Learning (MARL) and Federated Learning (FL) allows for experience sharing among agents while protecting users' privacy. Furthermore, Generative Adversarial Network (GAN) is used to generate state-action value distributions, improving the ability of traditional MARL methods to learn state-value information. By modeling the joint bandwidth and computing slicing resource management optimization problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), collaborative decision-making for slicing resource management is achieved by sharing only the generator network parameters of each agent through the combination of FL and GAN. This study provides a federated collaborative decision-making framework for addressing the slicing resource management problem in EC scenarios and offers theoretical support for enhancing the utilization efficiency of edge slicing resources while preserving users' privacy.

Methods The core concept of the proposed federated slicing resource management scheme is to first employ both GAN technology and the D3QN algorithm for local training within a multi-agent framework. The FL architecture is then used to share the generator network parameters of each agent, facilitating collaborative decision-making for joint bandwidth and computing slicing resource management. In this approach, each Access Point (AP) agent collects data on the total number of tasks to be transmitted and the number of Central Processing Unit (CPU) cycles required for computing tasks in each associated slice as local observations during each training time slot. Each agent subsequently selects the optimal local bandwidth and computing resource management action, obtaining the system reward, which consists of the average service waiting delay and service satisfaction rate, as well as the observation for the next time slot to train the local network. During the training process, each AP agent maintains its own main generator network, target generator network, and discriminator network. In each training episode, the D3QN algorithm is applied to decompose the state-action values, and GAN is used to perform multi-modal learning of the state value distribution, thus completing the local training. After each training episode, the AP agents upload their main generator network parameters for federated aggregation and receive the global main generator network parameters for the next training episode.

Results and Discussions By employing the D3QN algorithm and integrating the advantages of GAN within the MARL framework, alongside leveraging FL to share learning experiences among agents while protecting users' privacy, the proposed scheme reduces the long-term service waiting delay and improves the long-term average service satisfaction rate. Simulation results demonstrate that the proposed scheme achieves the highest average cumulative reward after approximately 500 episodes (Fig. 3), with a notable improvement of at least 10% in convergence performance compared to the baselines. Furthermore, the scheme strikes a better balance between average service waiting delay and average service satisfaction rate (Fig. 4). Additionally, it delivers superior performance in terms of user average service satisfaction rate, with at least an 8% improvement under varying user numbers (Fig. 5), highlighting its effectiveness in resource management under different task loads. Moreover, the proposed scheme reduces the average service waiting delay by at least 28% (Fig. 6) under varying numbers of agents.

Conclusions This paper investigates the joint bandwidth and computing slicing resource management problem in dynamic, unknown EC network scenarios and proposes a federated slicing resource management scheme based on GAN-assisted MARL. The proposed scheme enhances the agents' ability to learn state-value information and promotes collaborative learning by sharing the training network parameters of agents, which ultimately reduces long-term service waiting delays and improves long-term average service satisfaction rates, while protecting users' privacy. Simulation results show that: (1) The cumulative reward convergence performance of the proposed scheme improves by at least 10% compared to the baselines; (2) The average service satisfaction rate of the proposed scheme is more than 8% higher than that of the baselines under varying user numbers; (3) The average service waiting delay of the proposed scheme is reduced by at least 28% compared to the baselines under varying agent numbers. However, this study only considers ideal, static user scenarios and interference-free communication conditions. Future work should incorporate more real-world dynamics, such as time-varying user mobility and complex multi-user interference.

Key words: Edge Computing (EC); Network Slicing; Multi-Agent Reinforcement Learning (MARL); Federated Learning (FL); Generative Adversarial Network (GAN)