Computer Science and Application 计算机科学与应用, 2025, 15(5), 690-702
Published Online May 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.155141

Hans汉斯

低轨卫星无线接入网切片资源分配研究

黄禧龙，刘健培*，邝　坚

北京邮电大学计算机学院，北京

收稿日期：2025年4月22日；录用日期：2025年5月21日；发布日期：2025年5月28日

摘　要

随着低轨卫星网络规模扩大及其业务多样化发展，如何在动态网络环境下实现高效的切片资源分配成为亟待解决的关键问题。针对这一挑战，本文就低轨卫星无线接入网切片资源分配问题，提出一种基于模型强化学习的动态优化策略。本文将低轨卫星无线接入网切片资源分配问题建模成了一个控制问题。这个控制问题要求分配策略运行在的网络中在线学习，并且在学习过程中保持服务水平协议违约率低于阈值。在此基础上，本文提出了基于模型的强化学习的资源分配策略。策略包含一个基于核方法的分类器和一个用于控制模型预测错误率的模型自评估机制。实验结果表明，与其他强化学习策略相比，本文的策略在资源利用率、服务稳定性、平均执行时间等性能指标上均更优，更加适应低轨卫星网络环境。

关键词

低轨卫星，网络切片，强化学习

Research on Resource Allocation of Low Earth Orbit Satellite Radio Access Network Slice

Xilong Huang, Jianpei Liu*, Jian Kuang

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing

Received: Apr. 22nd, 2025; accepted: May 21st, 2025; published: May 28th, 2025

Abstract

With the rapid expansion of low Earth orbit (LEO) satellite networks and the diversification of their services, how to achieve efficient resource allocation for network slicing in dynamic networking environments has become a critical challenge. To address this problem, this paper proposes a model-

*通讯作者。

文章引用: 黄禧龙, 刘健培, 邝坚. 低轨卫星无线接入网切片资源分配研究[J]. 计算机科学与应用, 2025, 15(5): 690-702. DOI: 10.12677/csa.2025.155141

based reinforcement learning (MBRL) optimization strategy for radio access network (RAN) slicing resource allocation in LEO satellite networks. We formulate the RAN slicing resource allocation problem as a constrained control problem that requires: the allocation policy to perform online learning within the operational network, while maintaining the service level agreement (SLA) violation rate below a predefined threshold during the learning process. Building upon this formulation, we propose an MBRL-based resource allocation strategy featuring: a kernel method-based classifier for predictive modeling, and a model self-evaluation mechanism for error rate control. Experimental results demonstrate that compared with other reinforcement learning approaches, our strategy shows superior performance across multiple metrics including resource utilization efficiency, service stability, and average execution time, proving better adaptability to the unique characteristics of LEO satellite networks.

Keywords

LEO Satellite Network, Network Slice, Reinforcement Learning

1. 引言

LEO 卫星正在成为全球通信的重要补充，但其商业化落地仍面临着各种挑战。首先是业务多样性方面的挑战。随着网络与通信技术的不断发展，网络运营商往往需要在同一套物理网络设施上为各种用户提供多样化的服务，如支持宽带互联网、物联网、紧急通信等差异化服务[1]等。这些服务往往具有各异的、严格的服务质量(Quality of Service, QoS)需求。其中，LEO 卫星网络系统的服务级别协议(Service Level Agreement, SLA)保障能力尤为重要，是商业化落地的核心。运营商需向客户承诺带宽、延迟、可用性等指标，否则难以规模化推广。其次，LEO 卫星系统因其持续轨道运动导致星间连接状态频繁更迭，网络拓扑结构呈现出高度动态性，这不仅增加了空间环境复杂性，也对频谱等资源的分配和轨道规划提出了更严格的限制。同时，在应对不同垂直领域多样化业务需求时，该系统难以实现灵活化、定制化的服务供给，从而导致卫星资源的配置效率低下，整体利用效能受限。

四代移动通信向第五代的演进催生了网络切片技术[2]，这项技术通过逻辑隔离为不同业务提供定制化虚拟网络，是解决业务多样化的关键。但与地面网络不同，LEO 卫星系统的动态性导致了静态切片方案的不适用。

LEO 卫星网络系统涉及的资源很多，本文主要研究无线接入网络(Radio Access Network, RAN)切片的资源的分配。在现有研究中，一些关于地面网络 RAN 切片的研究考虑了最大化运营商长期利益的问题。其中一些采用基于模型的优化方法来解决问题[3] [4]。然而，随着网络规模的增加和缺乏有关服务的先验信息，经典的优化技术很难应对可重构 RAN 切片问题。随着人工智能(Artificial Intelligence, AI)技术的发展，一些论文将原始问题转化为马尔可夫决策过程(Markov Decision Process, MDP)，并使用深度强化学习(Deep Reinforcement Learning, DRL)算法来解决这些问题[5] [6]。除了服务需求波动外，由于卫星的移动性，卫星网络资源的可用性也是动态的。因此，地面网络的 RAN 切片策略不能直接用于卫星网络。有少数关于卫星 RAN 切片的研究[7] [8]，其中对资源进行切片以提高吞吐量。然而，LEO 卫星网络中的 RAN 切片的 SLA 波动控制问题尚未得到广泛研究。有研究者研究了天地一体化网络中的 RAN 资源切片和调度问题，提出了一种基于强化学习的两层联合资源切片和调度方案[9]。该方案通过分层决策机制，

有效应对了资源动态性和任务多样性的挑战。有研究者则在天地一体化网络中建立了低延迟、高吞吐量和广覆盖的RAN切片模型，设计了一种集中式和分布式多智能体DDPG (Deep Deterministic Policy Gradient)算法，以联合优化服务延迟、吞吐量和覆盖范围[10]。

上述研究在进行切片决策时，尚未同时考虑RAN切片的基本性能——资源利用率(Resource Utilization)和SLA满足率(SLA Satisfaction Ratio, SSR)。同时，先前研究的主要限制在于它们使用了无模型的强化学习(Model-free Reinforcement Learning, MFRL)方法，如果使用仿真系统或从真实环境获得的样本离线训练智能体(Agent)，这种方法会非常有效，但当智能体在真实的网络上学习时，这种方法并不特别合适。MFRL通常需要大量样本，这涉及对策略的广泛探索，包括低效策略。这可能会导致较长的训练时间，包含多次违反SLA的情况和额外的资源过度配置，这对租户和基础设施提供商都是不利的。最后，随着卫星网络规模的增大，资源管理变得更加复杂，需要为LEO卫星网络考虑合适的资源切片方案。

因此，本文计划在一个基于SDN的LEO卫星网络切片架构中，根据在线学习的思想，设计一个基于模型的强化学习(Model-based Reinforcement Learning, MBRL)的策略来应对上述问题和挑战。与其他基于MFBL的策略不同，MBRL通过构建环境模型来预测系统动态，从而减少对大量探索样本的依赖，提高学习效率和策略稳定性。本策略还包含一种自评估机制，控制环境模型出错概率。该策略可以更有效地管理卫星网络资源，优化资源利用率，同时确保SLA满足率，减少资源过度配置和SLA违约的风险。

2. 系统与问题建模

在本节中，本文首先描述了LEO卫星RAN的动态切片架构。然后，基于服务的特点，本文分析了eMBB (Enhanced Mobile Broadband)切片的资源利用率和SSR。最后，本文对切片的资源利用率和SSR的优化问题进行了建模和表示。

2.1. 切片架构与网络模型

本文基于图1所示的LEO卫星RAN资源分配的架构。本文考虑卫星与用户设备(User Equipment, UE)直接连接的场景，并将时间分为若干个切片窗口(Slice Window)。在每个切片窗口开始时，由地面的SDN控制器中的智能体负责分配切片间资源分配和切片编排。

在每个切片窗口内，地面SDN控制器中的智能体接收每个切片的观察向量，并且评估切片的SLA在上一决策阶段是否已实现。基于这些观察，智能体决定下一个切片窗口的各切片资源分配。控制目标是尽可能高效地分配资源块(Resource Block, RB)，同时确保SLA以高概率得到满足。

为了回避切片网络功能编排和切片内部资源分配问题，降低问题的复杂性，本文假设切片与卫星的关系为一对一，即每个切片编排中仅包含一颗卫星，每颗卫星也仅为一个切片提供其网络功能。同时，为了应对LEO网络拓扑结构的动态变化，本文假设在每个切片窗口内，卫星拓扑不变。在每个切片开始时，卫星网络拓扑结构发生变化。在本文的实验实现中，表现为卫星位置的随机变化。

我们假设用户存在于一个固定的小陆地区域内，并且卫星网络拓扑在切片窗口中保持不变。当LEO卫星与用户之间的仰角大于最小仰角时，LEO卫星可用，记为$l \in \mathcal{L}^w = \{1,2,\cdots,L^w\}$。因此，在切片窗口$w$中，当LEO卫星$l$的一个RB分配给用户$u$时，可达传输速率(Achievable Transmission Rate) $r_{l,u}^w$可以表示为公式1。

$$r_{l,u}^w = \log_2\left(1+\frac{PG\left(d_{l,u}^w\right)^{-\alpha}}{\sigma^2}\right), \tag{1}$$

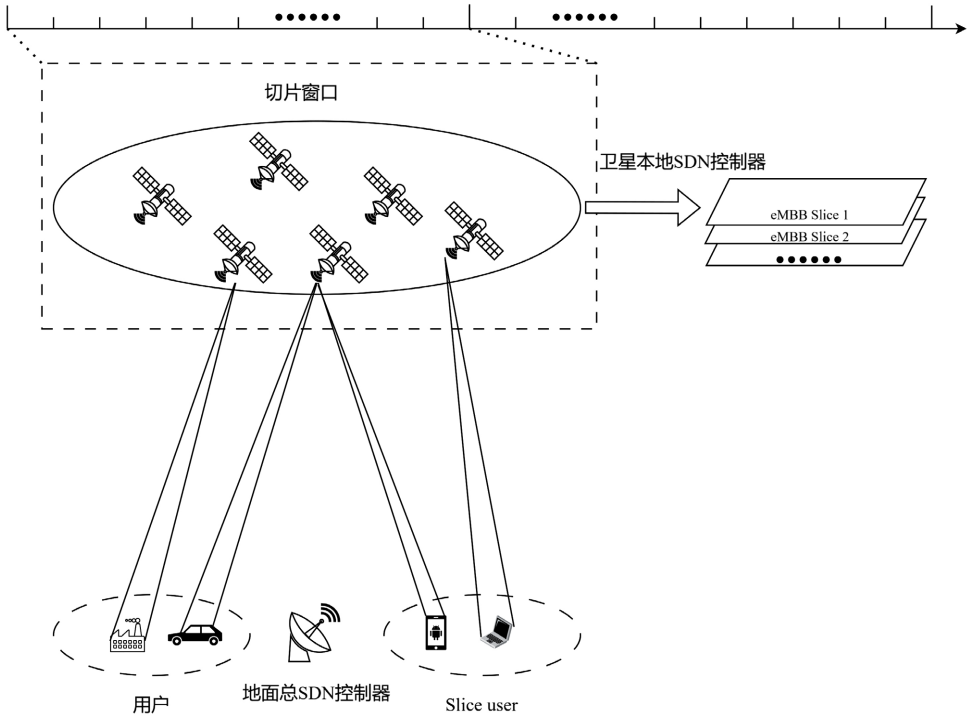

Figure 1. Diagram of satellite network slicing architecture based on SDN
图 1. 基于 SDN 的卫星网络切片架构示意图

其中，P 为传输功率(Transmit Power)，G 为常量功率增益因子(Power Gains Factor)，σ^2 为高斯噪声(Gaussian Noise Power)。考虑到大尺度衰落(Large-scale Fading)，传输功率也受到路径损耗指数(Path Loss Exponent) α 和卫星 l 与用户 u 之间距离的 $d_{l,u}^{w}$ 的影响。

在切片窗口 w 内的任意时间 t，一个用户同时最多可以访问一颗 LEO 卫星。当 LEO 卫星 l 能满足用户 u 的 SLA 需求的时，令 $I_{l,u}^{w}=0$，否则 $I_{l,u}^{w}=1$。因此，有公式 2 所示的约束条件。

$$\begin{aligned} &L^{w}-\sum_{l\in\mathcal{L}^{w}} I_{l,u}^{w}(t)\le 1,\forall w,u,t,\\ &I_{l,u}^{w}(t)=\{0,1\},\forall w,l,u,t. \end{aligned} \tag{2}$$

同时，在在切片窗口 w 内任意时间 t，卫星 l 分配给用户 u 的 RB 数量，不能超过自身被分配的数量总和。因此有公式 3 所示的约束条件。

$$\sum_{u\in\mathcal{U}^{w}} B_{l,u}^{w}(t)\le B_{l}^{w},\forall w,l,t. \tag{3}$$

$B_{l,u}^{w}$ 为用户被分配到的 RB 数量，B_{l}^{w} 为卫星被分配到的 RB 总量。

在本文中，eMBB 用户在一个切片窗口中会有一个固定的数据速率 R_{eMBB}，因此，RB 以静态的形式被分配给用户。因此，eMBB 切片的 SLA 可以表示为公式 4。

$$R_{l,u_e}^{w}\ge R_{eMBB},\forall w,l,u_e \tag{4}$$

为了达到这个需要，每个用户需要的 RB 可以表示为公式 5。

$$B_{l,u_e}^{w}=\left\lceil \frac{R_{eMBB}}{r_{l,u_e}^{w}} \right\rceil \tag{5}$$

由于大尺度衰落，传输功率随着距离衰减。因此，为了提高频谱效率，策略会优先选择距离最近且资源充足的 LEO 卫星连接用户；接着，根据用户的速率需求计算所需 RB 数量并进行分配，若当前卫星资源不足，则切换到下一个最近的卫星，卫星则记录当前切片窗口中被请求峰值 RB 数量中 p_n^i，用于判断切片在本切片窗口内是否满足 SLA 需求，直到所有卫星的资源均分配完毕或者所有用户的均得到满足。

2.2. 问题建模

为智能体寻找有效切片资源分配策略的问题可以表述为受约束的马尔可夫决策过程(Constrained Markov Decision Process, CMDP)，即公式 6。这个 CMDP 的目标寻找一个策略 $\pi \in \Pi$ 使得，分配的资源最少的同时，每个卫星 SLA 波动的数量的平均值控制在一个可接受的上界 δ。

$$\begin{aligned} &\min_{\pi\in\Pi} \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_\pi\left[\sum_{n=1}^{N}\sum_{i=1}^{K} A_n^{(i)}\right] \\ &\text{s.t. } \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_\pi\left[\sum_{n=1}^{N} I^{(i)}\left(S_n^{(i)}, A_n^{(i)}\right)\right] \le \delta, \text{ for all } i \in \mathcal{K} \end{aligned} \tag{6}$$

其中，$\mathcal{K}$ 表示切片总数，N 表示切片窗口数量。π 表示智能体决策，Π 表示智能体决策空间。$A_n^{(i)}$ 表示智能体对第 i 个切片在第 n 个切片窗口中所作的动作(Action)的所对应的随机变量。$S_n^{(i)}$ 表示智能体在第 n 个切片窗口结束时，观察到的第 i 个切片的观察向量 $s_n^{(i)}$ 的所对应的随机变量。$I^{(i)}\left(S_n^{(i)}, A_n^{(i)}\right)$ 用于指示切片 SLA 是否波动。$I^{(i)}\left(s_n^{(i)}, a_n^{(i)}\right)=0$ 表示在第 n 个切片窗口 SLA 稳定未波动，否则 $I^{(i)}\left(s_n^{(i)}, a_n^{(i)}\right)=1$。根据本文的具体场景，$s_n^{(i)}$ 包括拓扑结构变化后切片被请求的 RB 数量 b_{n+1}^i，和切片于上一个时间窗口中被请求的峰值 RB 数量 p_n^i。

由于系统动态未知且系统状态无法直接观察，因此无法直接解决此 CMDP 问题。即使没有这些限制，状态和控制空间的较大维度也会使常规数值方法变得不可行。

为了将 RL 应用于此问题，有必要将问题进行转化。常规的方法是将 SLA 指示函数加权到目标函数中。得到 MDP 问题为公式 7。

$$\min_{\pi\in\Pi} \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_\pi\left[\sum_{n=1}^{N}\sum_{i=1}^{K}\left[A_n^{(i)} + \lambda I^{(i)}\left(S_n^{(i)}, A_n^{(i)}\right)\right]\right] \tag{7}$$

$\sum_{i=1}^{K}\left[A_n^{(i)} + \lambda I^{(i)}\left(S_n^{(i)}, A_n^{(i)}\right)\right]$可以解释为系统在切片窗口 n 的代价，同时也可以表示为 $-r_n\left(\mathbf{s}_n, \mathbf{a}_n\right)$，即对于 $\mathbf{s}_n, \mathbf{a}_n$ 这组观察和动作的奖励的负数。很多 RL 算法都针对 discounted MDP。因此，可以进一步转化为公式 8。

$$\max_{\pi\in\Pi} \lim_{N\to\infty} \mathbb{E}_\pi\left[\sum_{n=1}^{N}\gamma^n r_n\left(\mathbf{S}_n, \mathbf{A}_n\right)\right] \tag{8}$$

其中，γ 是折扣因子，$\mathbf{S}_n, \mathbf{A}_n$ 表示系统轨迹在阶段 n 访问的随机观察-动作对。折扣 MDP 问题 8 在最大化期望回报，期望回报被定义为系统轨迹上折扣奖励的总和。

本文对于每一个切片，构建一个模型 $h_n^{(i)}$。该模型根据上一阶段结束时收到的观察值 $s_{n-1}^{(i)}$，预测特定的资源分配动作 $a_n^{(i)}$ 是否能够满足 SLA。在此基础上，我们可以将 CMDP 问题 6 转化为一个单步前瞻控制问题，从而得到一个模型预测控制器(Model Predictive Controller, MPC)。在这个控制器中，根据 $h_n^{(i)}$，观察-动作对必须满足每个切片的 SLA ($i \in \mathcal{K}$)。然而，这种方法没有考虑违反率界限 δ，预测器不够准确可能会导致过多的 SLA 违反情况。

因此，本文定义 $h_n^{(i)}$ 给出的$\left(s_n^{(i)}, a_n^{(i)}\right)$为 false negative 的概率为错误函数 $e^{(i)}$ 如公式 9。

$$e^{(i)}\left(h_n^{(i)}, s_{n-1}^{(i)}, a_n^{(i)}\right)=\mathbb{P}\left(I^{(i)}\left(S_n^{(i)}, a_n^{(i)}\right)=1 \mid h_n^{(i)}\left(s_{n-1}^{(i)}, a_n^{(i)}\right)=0\right) \tag{9}$$

按照惯例，零假设为在阶段 n 中，没有出现 SLA 违反情况，即 $I^{(i)}\left(S_n^{(i)}, a_n^{(i)}\right)=0$。因此，$e^{(i)}$ 表示第二类错误概率。

本文将问题转化为一个单步前瞻控制问题，其中每个控制动作 $\mathbf{a}_n$ 应该是合法的，并且 $\mathbf{a}_n$ 中的每个元素 $a_n^{(i)}$，根据 $h_n^{(i)}$ 都应该是能够满足 SLA 的，并且错误概率控制在界限 δ 以下。得到最终的问题建模为公式 10。

$$\begin{aligned}
&\min_{a_n^{(1)}\cdots a_n^{(i)}} \sum_{i=1}^{K} a_n^{(i)};\\
&\text{s.t. } e^{(i)}\left(h_n^{(i)}, s_{n-1}^{(i)}, a_n^{(i)}\right) \le \delta \quad \text{for } i \in \mathcal{K},\\
&\quad a_n^{(i)} \ge 0 \quad \text{for } i \in \mathcal{K},\\
&\quad \sum_{i\in\mathcal{K}} a_n^{(i)} \le C.
\end{aligned} \tag{10}$$

我们可以将上述问题分解成 K 个子问题，有助于在线操作。误差函数 $e^{(i)}$ 在线学习，辅助控制器做出决策。

3. 基于模型的强化学习资源分配策略

根据在线学习的思想。如图 2 所示，本文针对 LEO 卫星的 RAN 的基于模型的强化学习资源分配策略分为三个部分，分别是控制器(Controller)，假设学习器(H-learner)和误差学习器(E-learner)。其中，假设学习器中包含假设函数集，即 $\mathcal{H}_n=\left\{h_n^{(1)},\cdots,h_n^{(i)}\right\}$；误差学习器中包含误差函数集合假设，即 $\mathcal{E}_n=\left\{\hat{e}_n^{(1)},\cdots,\hat{e}_n^{(i)}\right\}$。其中，$n$ 表示第 n 个决策阶段。假设函数集负责预测系统对资源的分配动作是否能满足 SLA 需求；误差函数集负责预测不同安全裕度(Security Margin)之下的误差概率。控制器在每个决策阶段会根据上一个阶段的结束时观察到的系统状态以及假设学习器和误差学习器中的函数集，并且根据系统约束，得到最终的控制动作向量，即资源分配方案。

分配策略的模型的整体流程如算法 1 所示。在每个控制阶段，控制器都会根据假设函数集 $\mathcal{H}_n$ 与误差函数集 $\mathcal{E}_n$，并基于上一阶段结束时的观察状态 $\mathbf{s}_{n-1}$ 生成控制动作向量 $\mathbf{a}_n$，安全裕度向量 $\mathbf{m}_n$ 和各切片 SLA 违反预测向量 $\hat{\mathbf{y}}_n$。假设学习器会根据 $\mathbf{y}_n, \mathbf{s}_{n-1}, \mathbf{a}_n$ 更新假设函数集 $\mathcal{H}_n$；误差学习器会根据 $\hat{\mathbf{y}}_n, \mathbf{y}_n, \mathbf{m}_n$ 更新误差函数集 $\mathcal{E}_n$。

3.1. 假设学习器

H-learner 并行运行 K 个在线分类学习算法，每个算法对应 $\mathcal{H}_n$ 中的一个假设函数。这些函数在每个阶段都会根据 $\mathbf{S}_{n-1}, \mathbf{a}_n$ 和观察到的标签向量 $\mathbf{y}_n$ 进行更新和学习，并根据上一阶段的观察到的状态向量 $\mathbf{S}_n$，输出特定的控制动作向量 $\mathbf{a}_{n+1}$ 是否能满足切片的 SLA 需求的预测值 $\hat{\mathbf{y}}_n$。在本文中，假设函数的算法为 Projectron 算法[11]。

在决策阶段 $n+1$，假设学习器输入每个切片 $i \in \mathcal{K}$，上一个决策阶段，即决策阶段 n 时接收的系统状态向量 $\mathbf{s}_{n-1}^{(i)}$、上一个阶段的控制动作向量 $a_n^{(i)}$ 和观察到的结果 $y_n^{(i)}$。

为了进一步提升假设学习器对资源分配和 SLA 满足关系的学习能力，本文引入了样本增强策略。对于每个切片 $i \in \mathcal{K}$，如果 $y_n^{(i)}=0$，则表示决策阶段 n 到 $n+1$ 之间的切片的 SLA 已被满足；若 $y_n^{(i)}=1$，则表示未被满足。对于本文的具体场景而言，如果动作向量 a_n 使得 $y_n=0$，即决策阶段 n 分配的资源满足了切片的 SLA 需求的话，那么对于 $a'>a$，也可以认为其对应的结果为 $y_n=0$，即大于 a_n 的资源量，即 RB

数量，也可满足切片的资源需求；相对应地，如果 a_n 使得的 $y_n=1$，则对于 $a'<a$，也有 $y_n=1$。通过这种方式扩充样本，能增强假设函数对资源分配与 SLA 满足关系的学习能力，使模型能更好地适应网络切片的动态变化。因此，对于每个切片 $i\in\mathcal{K}$，当 $y_n=0$ 时，对所有大于其的值都进行更新；当 $y_n=1$ 时，对所有小于其的值都进行更新。

为了保证模型的稀疏性，在本文中，假设函数的算法为 Projectron 算法。Projectron 是一种特殊的基于核的感知机的算法(Kernel-based Perceptron)。Projectron 算法在模型的预测值与正确值不同时，会更新模型为 $h_n'=h_n+y_n\kappa(\mathbf{x}_n,\cdot)$，其中 $\mathbf{x}_n=(s_{n-1},a)$。同时，通过调整 $h_n''=\sum_{\mathbf{x}_n\in\mathcal{X}_n}\alpha_n\kappa(\mathbf{x}_n,\mathbf{x})$，中的系数 α_n，使得 h_n'' 最接近 h_n'。其中，$\mathcal{X}_n$ 为支持向量集，κ 为核函数。若 h_n'' 与 h_n' 的距离不大于一定的阈值 η，则不将 $\mathbf{x}_n$ 加入支持向量集，将 h_n 更新为 h_n''，否则将 $\mathbf{x}_n$ 加入支持向量集合，并且将 h_n 更新为 h_n'。Projectron 算法在每次更新时检查新加入的支持向量是否对分类结果有显著贡献，只有在满足一定的贡献条件时才将其纳入模型。这使得能够在降低计算复杂度的同时保持分类精度，有效适应动态变化的资源分配环境，从而提升模型的实时响应性和计算效率。

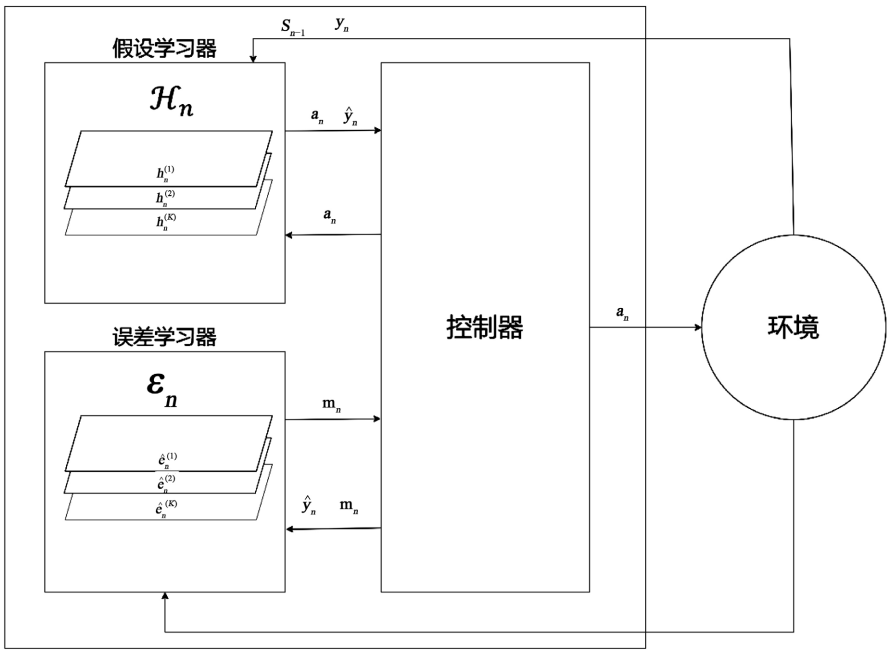

Figure 2. Diagram of model structure
图 2. 模型整体结构图

3.2. 误差学习器

误差学习器中并行运行 K 个估计函数 $\hat{e}_n^{(i)}(m)$，这些函数的输入包括当前预测值 $\hat{\mathbf{y}}_n$、真实观测值 $\mathbf{y}_n$ 和安全裕度 $\mathbf{m}_n$。从本文的实际场景和假设学习器中的样本增强策略出发，对于 $a<a'$，若有 $h_n^{(i)}\left(s_{n-1}^{(i)},a\right)=h_n^{(i)}\left(s_{n-1}^{(i)},a'\right)=0$ 的话，容易推导出，有 $e^{(i)}\left(h_n^{(i)},s_{n-1}^{(i)},a'\right)\le e^{(i)}\left(h_n^{(i)},s_{n-1}^{(i)},a\right)$，即动作控制向量分配的资源越多，假设函数犯第二类错误的概率越小。

$$m_n^{(i)}=a_n^{(i)}-\min\left\{a:h_n^{(i)}\left(s_{n-1}^{(i)},a\right)=0\right\} \tag{11}$$

基于上面结论，对于满足 $h_n^{(i)}\left(s_{n-1}^{(i)},a\right)=0$ 的最小的 a，我们定义动作控制向量 a' 的安全裕度为 $a'-a$，可以表示为公式 11。

算法 1 MBRL

输入：总资源量B，可靠性因子δ，$\mathbf{a}_{max}=\left(a_{max}^{(1)},\ldots,a_{max}^{(i)}\right)$

初始化：$\mathcal{H}_1=\{h_1^{(1)},\ldots,h_1^{(i)}\}$，$\mathcal{E}_1=\{\hat{e}_1^{(1)},\ldots,\hat{e}_n^{(i)}\}$

观察初始状态：获取初始观察状态$\mathbf{s}_0$

for $n=1,2,\ldots,N$ **do**

  for $i=1,2,\ldots,K$ **do** 控制器获取本轮预测结果

    $m_n^{(i)}\leftarrow min\{m:\hat{e}_n^{(i)}(m)\le\delta\}$ 根据假设函数获取安全裕度

    $a_n^{(i)}\leftarrow-1$，$\hat{y}_n^{(i)}\leftarrow1$

    while $\hat{y}_n^{(i)}=1$ and $a_n^{(i)}<a_{max}^{(i)}$ **do**

      $a_n^{(i)}=a_n^{(i)}+1$

      $\hat{y}_n^{(i)}\leftarrow h_n^{(i)}(s_{n-1}^{(i)},a_n^{(i)})$ 获取使得$h_n^{(i)}$预测 SLA 满足最小动作$a_n^{(i)}$

      if $\hat{y}_n^{(i)}=0$ **do**

        $m_n^{(i)}\leftarrow min(m_n^{(i)},a_{max}^{(i)}-a_n^{(i)})$ 根据最小动作更新安全裕度

        $a_n^{(i)}\leftarrow a_n^{(i)}+m_n^{(i)}$ 最小动作加上安全裕度得到新的动作

      end if

    end while

  end for

  if $\sum_i a_n^{(i)}>B$ **do**

    $\bar{a}^{(i)}\leftarrow\lfloor\frac{Ba^{(i)}}{\sum_{i'\in\mathcal{K}}a^{(i')}}\rfloor$ 使得最后的决策满足总资源量约束

    $m_n^{(i)}\leftarrow m_n^{(i)}-(a_n^{(i)}-\bar{a}^{(i)})$ 更新安全裕度

    $a_n^{(i)}\leftarrow\bar{a}^{(i)}$ 更新最终动作

    $\hat{y}_n^{(i)}\leftarrow h_n^{(i)}(s_{n-1}^{(i)},a_n^{(i)})$ 获取$h_n^{(i)}$对最终动作的预测

  end if

  地面控制器分配资源$\mathbf{a}_n$，并观察得到标签$\mathbf{y}_n$与状态$\mathbf{s}_n$

  $\mathcal{H}_{n+1}\leftarrow$ H-Leaner$(\mathcal{H}_n,\mathbf{y}_n,\mathbf{s}_{n-1},\mathbf{a}_n)$ 利用$\mathbf{y}_n,\mathbf{s}_{n-1},\mathbf{a}_n$ 更新假设函数

  $\mathcal{E}_{n+1}\leftarrow$ E-Leaner$(\mathcal{E}_n,\hat{\mathbf{y}}_n,\mathbf{y}_n,\mathbf{m}_n)$ 利用$\hat{\mathbf{y}}_n,\mathbf{y}_n,\mathbf{m}_n$更新误差函数

end for

使用函数$\hat{e}_n^{(i)}(m)$来估计在安全裕度m下的动作a'的假设函数分类错误概率$e^{(i)}\left(h_n^{(i)},s_{n-1}^{(i)},a'\right)$。因为$e^{(i)}$表示第二类错误的概率，因此函数$\hat{e}_n^{(i)}$仅在$\hat{y}_n^{(i)}=0$时更新。采用贝叶斯更新的方法更新误差函数估计值，其具体更新公式为 12。

$$\hat{e}_{n+1}^{(i)}(m)=(1-\beta)\hat{e}_n^{(i)}(m)+\beta y_n^{(i)} \tag{12}$$

其中β为学习率。在$\hat{y}_n^{(i)}=0$时，因为动作向量越大，即安全裕度越大，分类错误概率$e^{(i)}$越小，所以误差学习器也采用类似于假设学习器的样本增强策略，即当$y_n^{(i)}=0$时，当$y_n^{(i)}=0$时，对所有$m_n^{(i)}\le m\le a_{\max}^{(i)}$的$m$进行更新；当$y_n^{(i)}=1$时，对所有$0\le m\le m_n^{(i)}$的$m$进行更新。

3.3. 控制器

控制器是本文资源分配策略的核心模块，负责根据假设学习和误差学习器的输出生成资源分配决策。控制器的主要目标是保证切片的 SLA 要求得到满足的同时，最大化资源的利用效率，并避免资源的过度浪费。具体来说，在每个决策阶段 n，控制器通过上一阶段的系统状态 $\mathbf{s}_{n-1}$ 和假设学习器与误差学习器的输出，生成资源分配向量 $\mathbf{a}_n$。在此过程中，控制器不仅考虑每个切片的 SLA 满足情况，还根据误差学习器输出的安全裕度 $\mathbf{m}_n$ 对资源分配进行保守调整，从而确保 SLA 违约概率保持在可接受范围内。

为了实现这一目标，控制器需要满足全局约束条件。资源分配的总量不能超过系统的总容量 B，即必须满足 $\sum_{i=1}^{K} a_i \le B$。当系统决策与此发生冲突时，控制器会将决策投影到可行动作空间，即生成 $\overline{a}^{(i)}$ 使得 $\sum_{i\in K} \overline{a}_n^{(i)} = B$，且满足公式 13。

$$\frac{\overline{a}^{(i)}}{B} = \frac{a^{(i)}}{\sum_{i'\in K} a^{(i')}}. \tag{13}$$

通过这一系列步骤，控制器能够在保证切片 SLA 的基础上，优化资源的利用效率，同时提高对系统动态变化的适应能力。最终，控制器输出的资源分配向量 $\mathbf{a}_n$ 会被用于指导下一阶段的资源分配决策，并根据不断更新的状态、SLA 满足预测以及误差估计，逐步优化系统的整体资源分配策略。

4. 测试与评估

4.1. 实验设置

考虑到 LEO 网络切片系统的复杂性，本文没有采用仿真软件模拟星座一段时间的行动轨迹，同时据此模拟切片事宜的形式。而是仅利用 python 编码模拟，在每切片窗口结束的时候，组成切片的卫星位置会在切片中心点半径 30 km 范围之内随机变动，用来模拟卫星网络的动态性和变化性。本实验卫星高度为 350 km，卫星视场角度为 30 度。同时用户请求根据指数分布随机生成，运行了 1500 个切片窗口。本文其他实验参数入表 1 所示。

为了验证策略的性能，本文选取了以下四个主流强化学习算法进行对比。

(1) A2C (Advantage Actor-Critic) [12]：一种基于策略梯度的强化学习算法，结合了优势函数估计和价值函数逼近，以优化策略网络并提高学习效率。

(2) TD3 (Twin Delayed DDPG) [13]：双延迟深度确定性策略梯度算法，是对 DDPG 算法的改进，通过使用双 Q 网络、延迟策略更新和目标策略平滑来减少高估偏差，进而更稳定高效地学习连续动作空间中的最优策略。

(3) TRPO(Trust Region Policy Optimization) [14]：一种基于策略梯度的强化学习算法，通过限制策略更新的幅度来保证算法的稳定性，从而更有效地优化策略以实现长期累积奖励的最大化。

(4) SAC (Soft Actor-Critic) [15]：一种基于模型无关的无模型强化学习算法，结合了策略梯度和值函数估计，通过最大化累计奖励和最小化策略的熵来学习最优策略，以实现探索与利用的平衡。

本文主要从以下四个方面评估策略性能：

(1) 资源利用率：在本文的场景下，也为频谱效率。因为资源总量固定，所以资源利用率以每个切片窗口分配的资源数量之和表示。

(2) SLA 违约数：各个策略每个切片窗口的 SLA 违约次数。

(3) 累计 SLA 违约数：各个策略累计的 SLA 违约次数。

(4) 平均执行时间：策略每切片窗口平均执行时间。

Table 1. Experiment parameters
表 1. 实验参数表

实验参数	值
α	0.25
G	43.3 dBi
B	200
P	2 W
R_{eMBB}	10 Mbit/s
Bandwidth	180 KHz
β	0.01
η	0.1
δ	0.01
T_w	10s

4.2. 实验结果与分析

本实验的总分配 RB 数量，即频谱效率的结果图所图 3 所示。模拟实验共进行了 10 次，取平均值。从图 3 中可以得出，从策略的资源利用率角度出发，除去初始资源分配设定为的最大值的首个窗口之外，MBRL 策略分配的 RB 数量绝大部分窗口低于 100 RBs。其余四个策略中，A2C 算法和 TRPO 算法分配的 RB 数量相当，大部分窗口都在 150 RBs 以上；SAC 算法次之，有部分窗口分配 RB 数量在 150 RBs 以下；TD3 算法再次，但分配的 RB 数量仍大部分在 125RBs 以上，明显高于 MBRL 策略分配的数量。由此可以得出，在 5 个策略中，MBRL 策略有着最高的资源利用率，也即频谱效率，能够在更少的 RB 下满足资源需求，而其他策略则需要分配更多的 RB 来达到相似的效果。

从算法的分配结果的稳定性出发，TD3 策略对应的曲线波动最小，SAC 策略次之，A2C 和 TRPO 算法波动最大。MBRL 策略对应的曲线的波动介于 A2C、TRPO 和 SAC 之间，未出现过度波动，这表明 MBRL 在处理 LEO 卫星 RAN 切片网络动态资源分配时具备一定的稳定性。

从图 4 中可以看出 MBRL 策略对 SLA 的保证能力最强：除去最开始策略尚未充分学习系统的规律与特性的一个窗口有较多 SLA 违约次数之外，紫色曲线代表的 MBRL 策略在整个仿真过程中 SLA 违约次数非常低，基本维持在接近 0 的水平，说明 MBRL 能够很好地满足 SLA 的要求。这表明 MBRL 在资源分配和网络服务稳定性方面具有显著的优势。TD3 策略的 SLA 违约率最高：策略在切片窗口增多的情况下，SLA 违约次数显著上升，尤其在切片窗口达到 250 之后，违约次数急剧增加并在较高水平上波动。这说明 TD3 在满足 SLA 方面效果较差，尤其在资源需求或环境波动较大的情况下，其稳定性较低。其余三个策略虽然优于 TD3 策略，但 SLA 违约次数仍高于 MBRL 策略。从图 5 中可以看出，MBRL 在累积 SLA 违约方面表现最优，累积违约次数始终保持在较低水平，表明其在动态资源分配过程中能够有效减少长期的 SLA 偏差，实现较好的在线适应性和稳定性。相比之下，TD3 的累积 SLA 违约最多，尤其在切片窗口增加时呈现显著上升，表明其无法有效应对环境变化，对 SLA 的保障能力较弱。A2C、TRPO 和 SAC 的累积违约次数介于 MBRL 和 TD3 之间，表现中等，虽具备一定的 SLA 保障能力，但仍未能达到 MBRL 的稳定效果。整体来看，MBRL 在动态 LEO 卫星网络的资源分配中最能有效控制 SLA 违约率，在长时间内保持较优的 SLA 违约率。

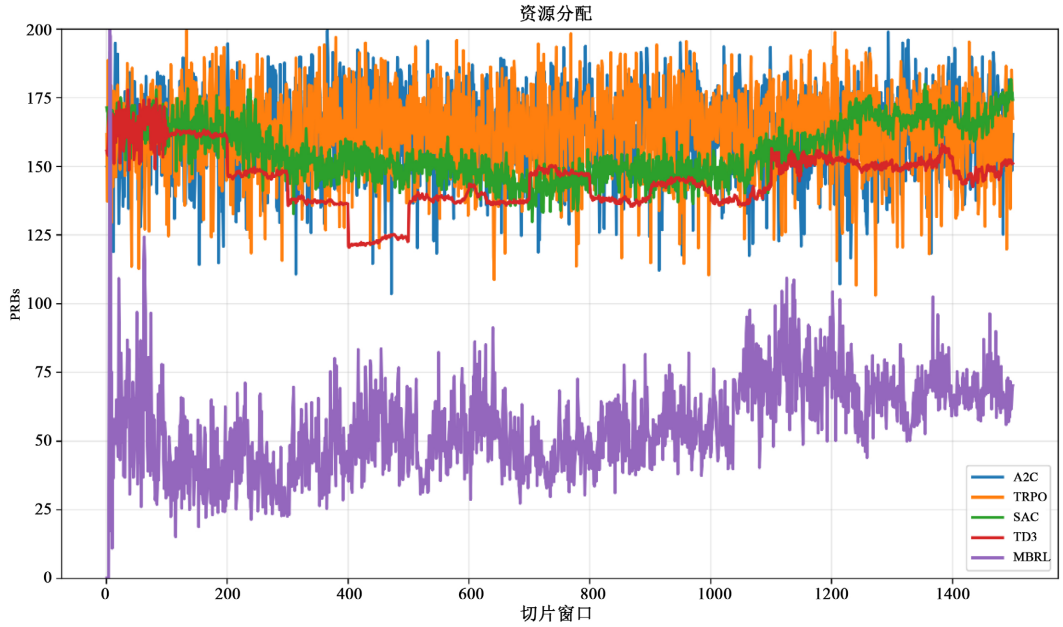

Figure 3. Results of total RBs allocated
图 3. 总分配 RB 数量结果图

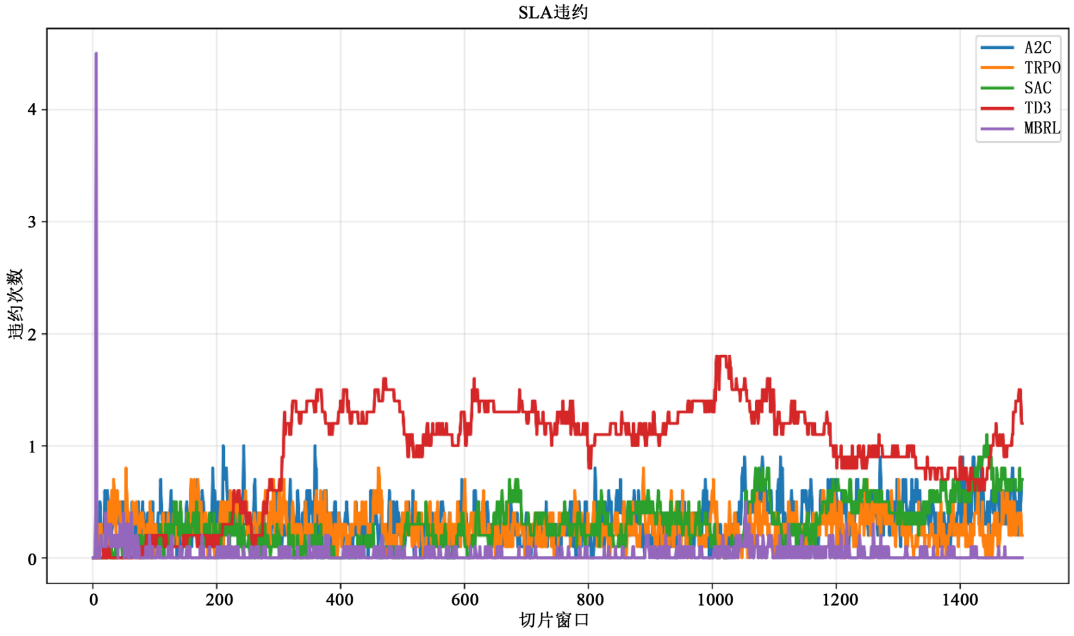

Figure 4. Results of SLA violations per stage
图 4. SLA 违约结果图

与 MFRL 的策略相比，MBRL 的策略通常有着更高的计算需求，这不利于计算资源受限的卫星网络环境。然而，本文采取的 Projectron 算法采用投影假设技术，避免支持集无界增长，使支持集规模在实验中保持较低水平。在 1500 步的实验后，支持集元素数量不多于 40 个，大幅降低了计算开销。从表 2 中可以得出，在训练过程中，MBRL 策略相较于其余四种策略有明显的优势。这方面也与策略的具体实现有密切关系。

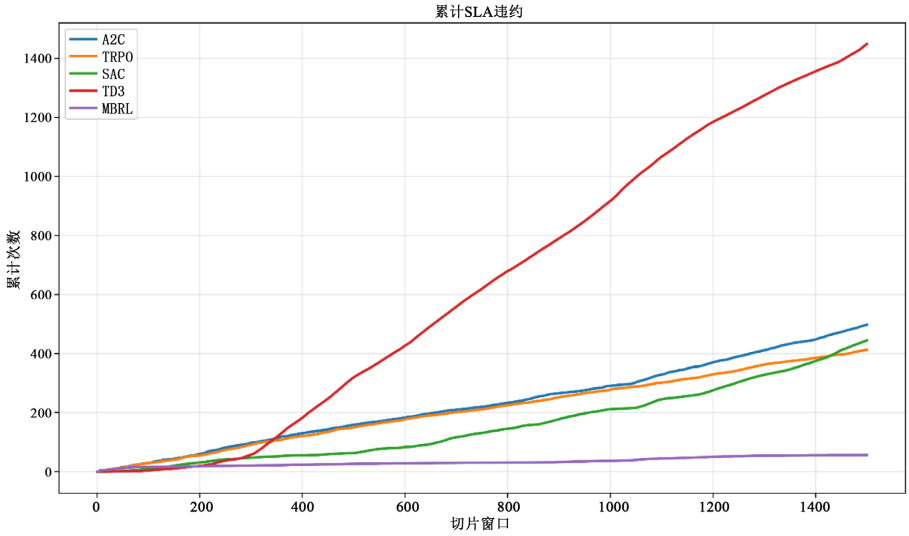

Figure 5. Results of cumulative SLA violations
图 5. 累计 SLA 违约结果图

实验结果表明，在动态 LEO 卫星网络的资源分配场景中，MBRL 策略在资源利用率、SLA 违约率、累积 SLA 违约方面和计算效率均表现最优。总体而言，MBRL 在 LEO 卫星 RAN 切片资源分配中展现了优越的资源分配稳定性、SLA 保障能力和计算效率，适合需要高服务质量保证的动态网络环境。

Table 2. Average execution time per slice window
表 2. 策略每切片窗口平均执行时间

策略名称	平均用时(ms)
A2C	153
TRPO	264
SAC	212
TD3	258
MBRL	77

5. 总结

本文针对 LEO 卫星 RAN 切片资源分配问题，在动态网络环境下提出了一种基于 MBRL 的动态优化策略。通过将问题建模为单步前瞻控制问题，要求分配策略在运行时进行在线学习，同时确保 SLA 违约率不超过设定阈值。所提策略创新性地引入了核方法分类器和模型自评估机制，以优化预测准确性并控制模型错误率。与传统强化学习方法相比，本策略在资源利用率、服务稳定性和执行效率等方面表现出显著优势，能够更好适应 LEO 卫星网络的动态环境，为 LEO 卫星网络的资源管理提供了有力的技术支持。

基金项目

本文由项目(D040304)资助。

参考文献

[1] Zou, C., Wang, H., Chang, J., Shao, F., Shang, L. and Li, G. (2022) Optimal Progressive Pitch for Oneweb Constellation

with Seamless Coverage. *Sensors*, **22**, Article No. 6302. https://doi.org/10.3390/s22166302

[2] 吴炀, 胡谷雨, 金凤林, 等. 卫星网络组网关键技术[J]. 指挥控制与仿真, 2022, 44(2): 88-100.

[3] Ko, H., Lee, J. and Pack, S. (2021) Priority-Based Dynamic Resource Allocation Scheme in Network Slicing. 2021 *International Conference on Information Networking* (*ICOIN*), Jeju Island, 13-16 January 2021, 62-64. https://doi.org/10.1109/icoin50884.2021.9333944

[4] Wang, Z., Wei, Y., Yu, F.R. and Han, Z. (2022) Utility Optimization for Resource Allocation in Multi-Access Edge Network Slicing: A Twin-Actor Deep Deterministic Policy Gradient Approach. *IEEE Transactions on Wireless Communications*, **21**, 5842-5856. https://doi.org/10.1109/twc.2022.3143949

[5] Jiang, M., Condoluci, M. and Mahmoodi, T. (2016) Network Slicing Management & Prioritization in 5G Mobile Systems. *European Wireless* 2016; 22*nd European Wireless Conference*, Paris, 11-12 October 2016, 1-6.

[6] Sun, S., Feng, X., Qin, S., Sun, Y. and Wang, G. (2020) Paired Bid-Based Double Auction Mechanism for RAN Slicing in 5G-and-Beyond System. 2020 *IEEE* 20*th International Conference on Communication Technology* (*ICCT*), Nanning, 28-31 October 2020, 533-538. https://doi.org/10.1109/icct50939.2020.9295796

[7] Yuan, S., Zhang, Y., Qie, W., Ma, T. and Li, S. (2021) Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network. *Computer Science and Information Systems*, **18**, 979-999. https://doi.org/10.2298/csis200710055y

[8] Wu, W., Dong, J., Sun, Y. and Yu, F.R. (2022) Heterogeneous Markov Decision Process Model for Joint Resource Allocation and Task Scheduling in Network Slicing Enabled Internet of Vehicles. *IEEE Wireless Communications Letters*, **11**, 1118-1122. https://doi.org/10.1109/lwc.2022.3152177

[9] Nassar, A. and Yilmaz, Y. (2022) Deep Reinforcement Learning for Adaptive Network Slicing in 5G for Intelligent Vehicular Systems and Smart Cities. *IEEE Internet of Things Journal*, **9**, 222-235. https://doi.org/10.1109/jiot.2021.3091674

[10] Wu, H., Chen, J., Zhou, C., Li, J. and Shen, X. (2021) Learning-Based Joint Resource Slicing and Scheduling in Space-Terrestrial Integrated Vehicular Networks. *Journal of Communications and Information Networks*, **6**, 208-223. https://doi.org/10.23919/jcin.2021.9549118

[11] Orabona, F., Keshet, J. and Caputo, B. (2009) Bounded Kernel-Based Online Learning. *Journal of Machine Learning Research*, **10**, 2643-2666.

[12] Mnih, V., Badia, A.P., Mirza, M., *et al*. (2016) Asynchronous Methods for Deep Reinforcement Learning. *International Conference on Machine Learning*, New York, 19-24 June 2016, 1928-1937.

[13] Fujimoto, S., Hoof, H. and Meger, D. (2018) Addressing Function Approximation Error in Actor-Critic Methods. *International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 1587-1596.

[14] Schulman, J., Levine, S., Abbeel, P., *et al*. (2015) Trust Region Policy Optimization. *International Conference on Machine Learning*, Lille, 7-9 July 2015, 1889-1897.

[15] Haarnoja, T., Zhou, A., Abbeel, P., *et al*. (2018) Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 1861-1870.