Software Engineering and Applications 软件工程与应用, 2023, 12(3), 530-542
Published Online June 2023 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2023.123052

Hans汉斯

基于分布式强化学习的功率控制算法研究

司　轲，李　烨

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2023年3月28日；录用日期：2023年6月22日；发布日期：2023年6月30日

摘　要

强化学习作为一种无模型的控制方法被应用于解决蜂窝网络中的同信道干扰问题。然而，在基于值的强化学习算法中，函数逼近存在误差导致Q值被高估，使算法收敛至次优策略而对信道干扰的抑制性能不佳，且在高频带场景中收敛速度缓慢。对此提出一种适用于分布式部署下的控制方法，使用DDQN学习离散策略，以添加三元组批评机制的延迟深度确定性策略梯度算法学习连续策略；使算法对动作价值的估计更准确，以提升算法在不同频带数量场景下对干扰的抑制性能。通过数量的扩展性实验表明了所提算法在不同频带数量场景下，保证更快收敛速度的同时对信道干扰有更好的抑制效果，证明了算法的有效性与扩展性。

关键词

分布式强化学习，功率控制，Actor-Critic算法，双重深度Q网络，延迟深度确定性策略梯度

Research on Power Control Algorithm Based on Distributed Reinforcement Learning

Ke Si, Ye Li

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 28th, 2023; accepted: Jun. 22nd, 2023; published: Jun. 30th, 2023

Abstract

Reinforcement learning is applied as a model free control method to solve the problem of co channel interference in cellular networks. However, in value based reinforcement learning algorithms, error in function approximation leads to overestimation of the Q value, which leads to the algorithm converging to a suboptimal strategy and poor performance in suppressing channel interference, and the convergence speed is slow in high-frequency scenarios. This paper proposes a control method suitable for distributed deployment, which uses DDQN to learn discrete strategies, and adds a delay-depth deterministic strategy gradient algorithm with a triplet criticism mechan-

文章引用: 司轲, 李烨. 基于分布式强化学习的功率控制算法研究[J]. 软件工程与应用, 2023, 12(3): 530-542.
DOI: 10.12677/sea.2023.123052

ism to learn continuous strategies; Make the algorithm's estimation of action value more accurate to improve the algorithm's interference suppression performance under different frequency band number scenarios. Quantitative scalability experiments have shown that the proposed algorithm guarantees faster convergence speed and better suppression of channel interference in different frequency band scenarios, demonstrating the effectiveness and scalability of the algorithm.

Keywords

Distributed Reinforcement Learning, Power Control, Actor-Critic Algorithm, Dual Depth Q Network, Delay Depth Deterministic Strategy Gradient

1. 引言

随着蜂窝通信设备数量的不断增长，对设备间的干扰抑制，已成为部署大型网络的关键问题。小区基站对其覆盖范围内的某一设备提高发射功率，可增加该设备与基站间链路的频谱效率，但同时也会对其它设备的链路产生干扰。而不同频带上的信号间彼此相互正交，不会产生干扰[1]。因此频分多址系统被广泛应用于蜂窝通信中，以抑制设备间的干扰。但当多个链路共享同一个频带时，联合频带选择的功率控制被认为是 NP-HARD [2]。

对于该问题，已有集中式的迭代方法被应用于多频带场景，以使链路平均频谱效率达到最优结果。虽然集中式的迭代算法在仿真和理论分析中表现出优异的性能，但它们在实际系统中的实现面临严重的障碍。由于信道状态是随时间演化的，此类算法通常是以相当大的计算复杂度为代价[3]，以迭代的方式找到一个近似最优的方案，例如加权最小均方误差算法[4] (WMMSE)需要在每次迭代中进行复杂的运算，在信道快速变化的实际蜂窝通信场景下实现完美的频带、功率分配，仍然是一个挑战[5]。

为此，业界对于无模型驱动的强化学习方法展开了研究。文献[6]使用单一 Q 网络同时学习频带选择与功率控制策略，但该方法的动作空间是可选频带数量的笛卡尔积，随频带数增多而难以收敛，且需对连续功率域做量化[7]，引入量化误差。文献[8]提出了一种分层控制策略，上层使用 DQN 学习离散动作，下层使用 DDPG 学习连续功率分配动作，解决了单一 Q 网络无法同时处理离散与连续策略的问题，使算法具有同类算法中的最优性能。然而，DQN 与 DDPG 为基于值的强化学习算法，包含了一个过估计动作值的最大化步骤，非均匀的高估使得算法学习到不切实际的高动作值。文献[9]指出，当动作值不准确时，无论函数逼近误差来源如何，都会产生过高估计，这将导致算法渐进收敛至次优策略。对此，文献[10]提出使用双 Critic 网络来估算网络目标，并在连续控制任务上取得了更优的性能。然而，文献[11]指出，以多个网络估算较小值的方法会产生低估动作值的问题，并从理论上证明了其存在性。

为此，本文设计了一种新的分层控制策略，以双重深度 Q 网络(Double Deep Q Network, DDQN)学习离散子频带选择策略，以双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient, TD3)处理连续功率控制策略，并在此基础上，引入三元组批评机制，提出三延迟深度确定性策略梯度的控制方法：采用三套 Critic 网络，使用其中两个 Critic 网络估算较小值，第三个 Critic 网络单独估算动作值，并以权重结合两项共同计算目标值，有效缓解了不准确动作值造成的 q 值高估问题，提升了网络的性能。实验结果表明，相同部署下，本文所提算法拥有最高的链路平均频谱效率，且随频带数增多拥有更快的收敛速度。

2. 系统模型

本文考虑一个具有 K 个小区的蜂窝网络，基站作为分布式的智能体位于各小区中心。网络中共有 $N=\{1,\cdots,n,\cdots,N\}$ 条链路，通过共享 $M=\{1,\cdots,m,\cdots,M\}$ 个频带传输信息。链路 n 由设备 n 与关联的小区基站共同组成。设该网络为一个持续时长为 T 的完全同步的时隙系统，且所有发射机与基站都配备单根天线。由于实际场景下频带的稀缺性，设 $K \gg M$ ，每条链路在时隙开始时选取一个频带[12]。

本文的信道模型由两部分组成：多径衰落与阴影衰落。设阴影衰落在所有频带上均为相同的，多径衰落具有频率选择性，在每个频带上，多径衰落是块衰

落与平坦的。设在时隙 t、选取频带 m 时，设备 n 到基站的下行链路信道增益为：

$$g_{s\to r,m}^{(t)}=\left|h_{s\to r,m}^{(t)}\right|^2\cdot\alpha_{s\to r}^{(t)} \tag{1}$$

其中， $h_{s\to r,m}^{(t)}$ 、 $\alpha_{s\to r}^{(t)}$ 分别为设备 s 到基站 r、选取频带 m 时的多径衰落[13]与阴影衰落。使用 Jakes 衰落模型[14]描述 $h_{s\to r,m}^{(t)}$ ：

$$h_{s\to r,m}^{(t)}=\rho_r^{(t)}\cdot h_{s\to r,m}^{(t-1)}+\sqrt{1-\rho_r^2}\cdot e_{s\to r,m}^{(t)} \tag{2}$$

该模型作为一阶复高斯马尔可夫过程引入，其中 ρ_r 为零阶贝塞尔函数，表示两个连续衰落块间的相关性：

$$\rho_r^{(t)}=J_0\left(2\pi\cdot f_{d,r}^{(t)}\cdot T\right) \tag{3}$$

其值取决于最大多普勒频率 $f_{d,r}^{(t)}$ 。表示为：

$$f_{d,r}^{(t)}=\frac{V_r^{(t)}\cdot f_c}{c} \tag{4}$$

其中 $V_r^{(t)}$ 为设备的移动速度、 f_c 为载波频率、 c 为真空光速。 $e_{s\to r,m}^{(t)}$ 是具有单位方差的独立同分布复高斯变量，信道根据 $e_{s\to r,m}^{(t)}$ 在蜂窝小区中更新。

使用二进制变量 $\xi_{n,m}^{(t)}$ 表示链路 n 对频带 m 的选取。设基站在时隙 t 对链路 $c_n^{(t)}$ 的发射功率为 $p_n^{(t)}$ ，则设备 n 在时隙 t、频带 m 上的信噪比为：

$$\gamma_{n,m}^{(t)}\left(c^{(t)},p^{(t)}\right)=\frac{\xi_{n,m}^{(t)}\cdot g_{c_n^{(t)}\to n,m}^{(t)}\cdot p_n^{(t)}}{\sum_{q\neq n}\xi_{q,m}^{(t)}\cdot g_{c_q^{(t)}\to n,m}^{(t)}\cdot p_q^{(t)}+\sigma^2} \tag{5}$$

σ^2 为加性高斯白噪声功率谱密度，为一常数。式中分子项为时隙 t 中链路 $c_n^{(t)}$ 选取频带 m 时产生的下行链路信道增益，分母项为同时刻小区中其他链路选取频带 m 时，对 $c_n^{(t)}$ 产生的下行信道干扰。

假设归一化带宽，时隙 t 下链路 $c_n^{(t)}$ 选取频带 m 时的信道增益为：

$$C_{c_n^{(t)},m}^{(t)}=\log\left(1+\gamma_{n,m}^{(t)}\left(c^{(t)},p^{(t)}\right)\right) \tag{6}$$

对于给定链路 $c_n^{(t)}$ ，和速率最大化问题可表述为：

$$\begin{aligned}\underset{p^{(t)},\xi^{(t)}}{\text{maximize}}\quad & \sum_{n=1}^{N}\sum_{m=1}^{M}C_{n,m}^{(t)}\\ \text{subject to}\quad & 0\le p_n^{(t)}\le P_{\max},\forall n\in N\\ & \xi_{n,m}^{(t)}\in\{0,1\},\forall n\in N,\forall m\in M\\ & \sum_{m\in M}\xi_{n,m}^{(t)}=1,\forall n\in N\end{aligned} \tag{7}$$

由于设备移动导致阴影衰落与多径衰落涉及时变性，即使对于给定频带分配方案 $\xi_{n,m}^{(t)}$，上述问题也被证明为 NP-HARD [15]。

3. 深度强化学习功率控制方法

强化学习是一种学习如何从状态映射到行为以使得获取的累计折扣奖励最大化的机制。智能体需不断地在环境中进行探索，通过环境给予的奖励来不断优化状态－动作的对应关系[16]。设智能体在时隙 t 下通过对本地环境的观察得到的状态为 $s_t \in S$，智能体根据策略 $\pi(\theta): S \to A$ 选择动作 $a^{(t)} \in A$，使其根据状态转移概率 p 进入下一状态 $s^{(t+1)} \in S$，同时获得环境对智能体执行动作 $a^{(t)}$ 的奖励 $r^{(t)}$。将四元组 $e^{(t)} = \left(s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)}\right)$ 称为智能体在时隙 t 下获得的经验，强化学习的目标是寻找最优策略 $\pi(\theta)$，最大化从任意状态－动作对始的期望累计折扣奖励：

$$E_{s^{(t)},a^{(t)}\sim\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t \cdot r\left(s^{(t)}, a^{(t)}\right)\right] \tag{8}$$

其中 γ 为折扣因子。设智能体在任意时刻 t 下通过观测环境获取完整的状态，则上述学习过程可被描述为马尔可夫决策过程。

对于处理连续控制任务的 Actor-critic 方法中，策略 π_ϕ 可以通过期望累积折扣奖励的梯度 $\nabla_\phi E(\phi)$ 更新：

$$\nabla_\phi E(\phi) = E_{s^{(t)},a^{(t)}\sim\pi^*}\left[\nabla_a Q^\pi\left(s^{(t)}, a^{(t)}\right)|_{a^{(t)}} \nabla_\phi \pi_\phi(s)\right] \tag{9}$$

其中 $Q^\pi(s,a) = E_{s^{(t)},a^{(t)}\sim\pi}\left[R_t \mid s^{(t)}, a^{(t)}\right]$，为智能体在状态 $s^{(t)}$ 下执行动作 $a^{(t)}$ 所获得的期望奖励，称为值函数。

对于处理离散控制任务的 Q 学习中，值函数可以根据下一时刻的状态－动作对，通过基于贝尔曼方程的时间差分学习[17]表示为：

$$Q^\pi\left(s^{(t)}, a^{(t)}\right) = r + \gamma * E_{s^{(t+1)},a^{(t+1)}}\left[Q^\pi\left(s^{(t+1)}, a^{(t+1)}\right)\right], a^{(t+1)} \sim \pi\left(s^{(t+1)}\right) \tag{10}$$

值函数可以使用参数为 θ 的可微函数近似值 $Q_\theta(s,a)$ 进行估算，通过使用固定目标网络的时间差分技术来更新网络，以在多次更新中维持固定的目标值 y：

$$y = r + \gamma * Q_{\theta'}\left(s^{(t+1)}, a^{(t+1)}\right), a^{(t+1)} \sim \pi_{\phi'}\left(s^{(t+1)}\right) \tag{11}$$

本文将各个小区基站设置为独立的智能体，负责收集其本地通信范围内状态信息、经验数据，并负责在链路间传输频带选取与功率分配动作以完成控制。所有智能体共享相同的网络参数。由于分布式智能体的设置违反了马尔可夫假设，通过收集各个智能体在同一时隙下的经验数据，存储在一固定容量的经验回放池中均匀抽样学习以确保稳定性。

3.1. 强化学习函数逼近误差问题

在具有离散动作的 Q 学习中，通过对式(11)进行贪婪的值估计，当目标值受到误差为 ε 的影响时，以目标值加上误差 ε 的最大化过程通常大于真实估算值，即：

$$E_\varepsilon\left[\max_{a^{(t+1)}}\left(Q\left(s^{(t+1)}, a^{(t+1)}\right) + \varepsilon\right)\right] \geq \max_{a^{(t+1)}} Q\left(s^{(t+1)}, a^{(t+1)}\right) \tag{12}$$

这会导致算法在值更新过程中产生高估偏差，该偏差会通过贝尔曼方程进行传播。由于动作的真实价值在强化学习训练过程中不可知，因此这种函数近似引起的误差是不可避免的[18]。

过估计问题也存在于通过梯度下降方法更新连续动作的 Actor-critic 算法中。不准确的价值估计会导致策略更新不力，这会导致网络创建一个错误的反馈循环，其中次优动作会被次优 Critic 高度评价，从而在下一次更新中强化对次优动作的选取。本文针对文献[8]提出的使用 DQN 学习离散策略、以 DDPG 学习连续策略中出现的 Q 值高估问题，以更先进的算法替代以降低函数逼近误差的影响。

3.2. 双重 Q 网络

DQN 算法采用 $\max_a q\left(s_{t+1},a\right)$ 来更新动作价值，这样会导致最大化偏差[18]。为了解决该问题，DDQN 算法使用两个独立的动作价值估计值：$q_{1,2}\left(s^{(t+1)},\arg\max_a q_{1,2}\left(s^{(t+1)},a\right)\right)$ 代替 $\max_a q\left(s^{(t+1)},a\right)$ 过程。因此有：

$$E\left[q_1\left(s^{(t+1)},A^*\right)\right]=q\left(s^{(t+1)},A^*\right)A^*=\arg\max_a q_2\left(s^{(s+1)},a\right) \tag{13}$$

由于 q_1 与 q_2 是相互独立的估计，借此消除了偏差。在 DDQN 学习过程中，q_1 与 q_2 都需要逐步更新。DDQN 算法相较于 DQN 算法，主要针对后者的过估计问题，改变了目标值的计算方法，其他地方与 DQN 算法完全一致，DDQN 算法流程框图如图 1 所示：

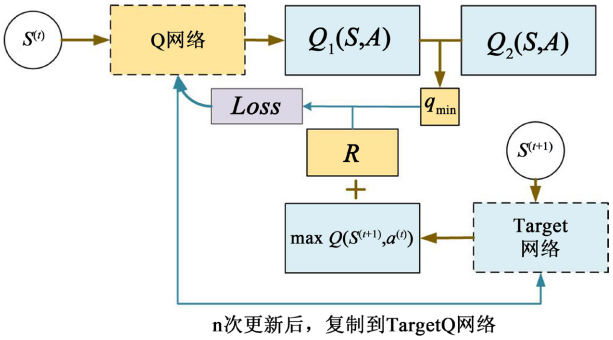


Figure 1. Dual Q network parameter learning mechanism
图 1. 双重 Q 网络参数学习机制

3.3. 改进双延迟深度确定性策略梯度

TD3 算法需用到 6 个网络，网络结构如图 2 所示：

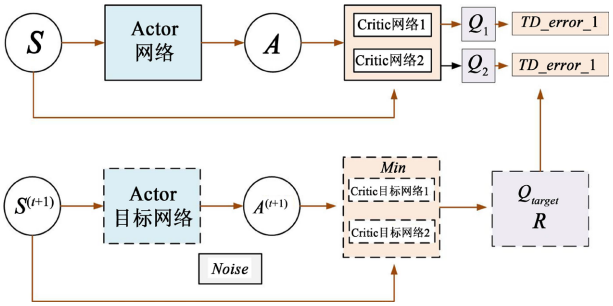


Figure 2. Learning mechanism of dual delay depth deterministic strategy gradient algorithm
图 2. 双延迟深度确定性策略梯度算法学习机制

相比于文献[8]中使用的 DDPG 算法，TD3 使用两套 Critic 网络估算 Q 值，并使用较小项对真实 Q 值进行估算。当使用两个独立 Critic 网络估算目标 Q 值时，它们可以被用来对选择的动作进行无偏估计

[10]。因此，TD3 的主网络有 2 个 Critic 网络和一个 Actor 网络，同时目标网络也有主网络的一个备份。通过增加一个 Critic 网络，在多个 Critic 网络间形成对比，选取最小的 q 值，来避免持续过高的估计。

此外，TD3 相比 DDPG 算法使用了延迟更新策略。相比 DDPG 中 Actor 网络参数随 Critic 网络参数相应更新，采用延迟更新策略的 Actor 网络参数将在 Critic 网络更新多次后再进行更新，通过减少错误更新的方式来稳定 Q 值。

TD3 算法还通过添加随机噪声的方式更新 Critic 网络，以达到对 Critic 网络波动的稳定性。通过在下一个状态的动作上加入扰动以计算目标值，平滑目标策略，使价值评估更准确。因此，TD3 算法中 Critic 网络参数更新方式为：

$$\begin{aligned} &\tilde{a}=\pi_{\theta'}\left(s^{(t+1)}\right)+\xi,\xi\sim clip\left(N\left(0,\delta^2\right),-c,c\right)\\ &y=r+\gamma*\min_{i=1,2}Q_{\theta_i'}\left(s^{(t+1)},\tilde{a}\right)\\ &\theta_i=\arg\min_{\theta_i}N^{-1}\sum\left(y-Q_{\theta_i}\left(s^{(t)},a^{(t)}\right)\right)^2 \end{aligned} \tag{14}$$

其中 $\tilde{a}$ 为 Critic 的输出动作，ζ 为剪切后的噪声，服从方差为 δ^2 的正态分布，θ_i 代表多个 Critic 网络。

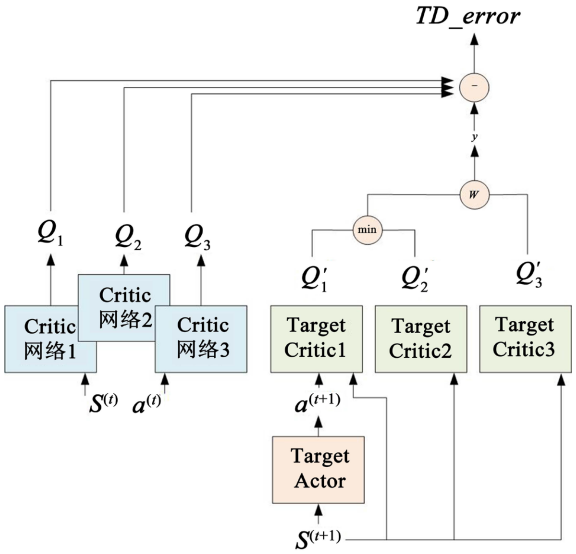


Figure 3. Triple critic weighted action value update strategy
图 3. 三元批评者加权动作值更新策略

然而，TD3 算法通过设立 2 个 Critic 网络同步独立估计，并取最小值估算 Q 值的方法在消除连续控制问题中的高估现象方面是有效的，但在每次更新迭代时，不够灵活的函数近似使得算法对后续状态的估计仍不精准，使算法根据时间差分进行的学习产生了低估偏差[11]。尽管这种偏差不会在更新过程中明确传播[16]，但不准确的估计仍然会对性能产生负面影响。文献[11]提出了一种解决方案用于限制 TD3 中的低估现象。为此，本文采用三元组批评者加权动作值方法进行更新。

如图 3 所示，本文用双 Critic 网络 θ_1、θ_2 选择最小估计值来解决 DDPG 中的高估问题，并添加单个 Critic 网络 θ_3 来解决 TD3 算法中的低估问题，以权重结合这两种相反的偏差，以实现更准确的估计。

3.4. 强化学习设计

为了允许分布式执行，智能体在本地通信环境下收集链路观测信息时，通过将其它链路信息视为其

本地环境的一部分，作为独立的智能体对环境进行观测。强化学习各部分的设计如下：

1) 动作集设计

所有分布式执行的智能体都有相同的动作空间，其中下层使用由可选子频带数量为界组成的离散动作空间，即：$a_n^{(t)} \in A_M = \{1,\cdots,m,\cdots,M\}$，本文定义智能体在时隙 t 下的子频带选取动作为 $a_n^{(t)}$。上层使用一个连续动作空间，定义为：$A_P \in [0,1]$，上层网络是在下层之后运行的，因此设上层输出动作为 $a_{n,a_n^{(t)}}^{(t)}$。对于某一确定输出动作，连续功率输出可表示为：

$$p_n^{(t)} = P_{\max} * a_{n,a_n^{(t)}}^{(t)} \tag{15}$$

2) 状态集设计

状态集表示分布式部署下的智能体对本地无线环境的观察状况。对于链路 n_1，蜂窝小区中其它链路 n_2 $(n_1,n_2 \in N,\ n_1 \neq n_2)$根据在时隙 t、子频带 m 上的下行链路信道增益按照降序排列，取前 k 项作为集合 $U_{n,m}^{(t)}$，以表示小区中其余链路的干扰。则在时隙 t、子频带 m 上，对于每个分布式部署下的智能体 n，离散策略状态集 $S_{n,m}^{(t)}$ 内容包括：发射功率大小 $\alpha_{n,m}^{(t-1)} * p_n^{(t-1)}$、下行链路信道增益 $g_{n,m}^{(t-1)}$、对网络总和目标的贡献 $C_n^{(t-1)}$、以及其它链路对本地链路的干扰 $\sum_{n_i \neq n} \alpha_{n_i}^{(t-1)} * p_{n_i}^{(t-1)}$、$g_{n_i,m}^{(t-1)}$。由于上层执行需要来自所有子频带信息的下层离散策略，因此上层需要有比下层更高维的环境观测。因此，连续策略状态集为：$S_n^{(t)} = \left\{S_{n,1}^{(t)},\cdots,S_{n,m}^{(t)},\cdots,S_{n,M}^{(t)}\right\}$。

3) 奖励函数设计

两个学习层联合学习旨在最大化式(7)中小区间链路总和频谱效率，因此应共享相同的奖励函数。奖励函数描述了分布式部署的智能体依次执行两个动作后对目标的整体贡献。设智能体 n 在时隙 t、子频带 m 上对其他链路的干扰惩罚项为 $\pi_{n\to n_k}^{(t)}$：

$$\pi_{n\to n_k}^{(t)} = \log\left(1 + \frac{\alpha_{n_k,a_n^{(t)}}^{(t)} * g_{n_k\to n_k,a_n^{(t)}}^{(t)} * p_{n_k}^{(t)}}{\sum_{n_i \neq n,n_k} \alpha_{n_i,a_n^{(t)}}^{(t)} * g_{n_i\to n_k,a_n^{(t)}}^{(t)} * p_{n_k}^{(t)} + \sigma^2}\right) \tag{16}$$

则奖励函数 $r_n^{(t+1)}$ 被定义为：

$$r_n^{(t+1)} = C_{n,a_n^{(t)}}^{(t)} - \sum_{n_k \in U_{n,a_n^{(t)}}^{(t+1)}} \pi_{n\to n_k}^{(t)} \tag{17}$$

3.5. 联合控制方案

本文提出的分层策略算法流程图如图 4 所示。

在每个训练时隙 t 开始时，每个分布式部署下的智能体 n，将本地无线环境观察 $s_n^{(t)}$ 输入下层网络 (ϕ_{agent})，输出子频带选择动作 $a_n^{(t)}$，同时将累积经验 $e^{(t)}$ 存入经验回放池中存储。中心化训练智能体在下层网络创建参数为 ϕ_{target} 的目标网络，并通过两个独立网络$(\phi_1,\ \phi_2)$同步估算 q 值，并取较小值作为 q_{choose}，通过梯度下降：

$$\nabla_\phi \frac{1}{|B|} \sum_e \left(y\left(r_n^{(t+1)},s_n^{(t+1)}\right) - q_{choose}\left(s_n^{(t)},a_n^{(t)},\phi_{1,2}\right)\right)^2 \tag{18}$$

以最小化贝尔曼误差估算动作真实价值，其中 $y\left(r_n^{(t+1)},s_n^{(t+1)}\right) = r_n^{(t+1)} + \gamma * \max q_{choose}\left(s_n^{(t)},a_n^{(t)}\right)$。

根据下层输出的子频带选择动作 $a_n^{(t)}$，分布式执行下的智能体 n 的上层网络生成状态 $s_{n,a_n^{(t)}}^{(t)}$ 输入网络 θ_{agent}，网络 θ_{agent} 输出功率分配动作 $a_{n,a_n^{(t)}}^{(t)}$，并将此时生成的经验 $e^{(t)}$ 存入经验回放池中存储以待中心化训练算法抽样学习。中心化训练智能体的上层在此时创建参数为 $\zeta_i\ (i=1,2,3)$ 的 Critic 网络同步估算动作真实价值，并以权重组合作为 q'_{choose} 输出，以梯度：

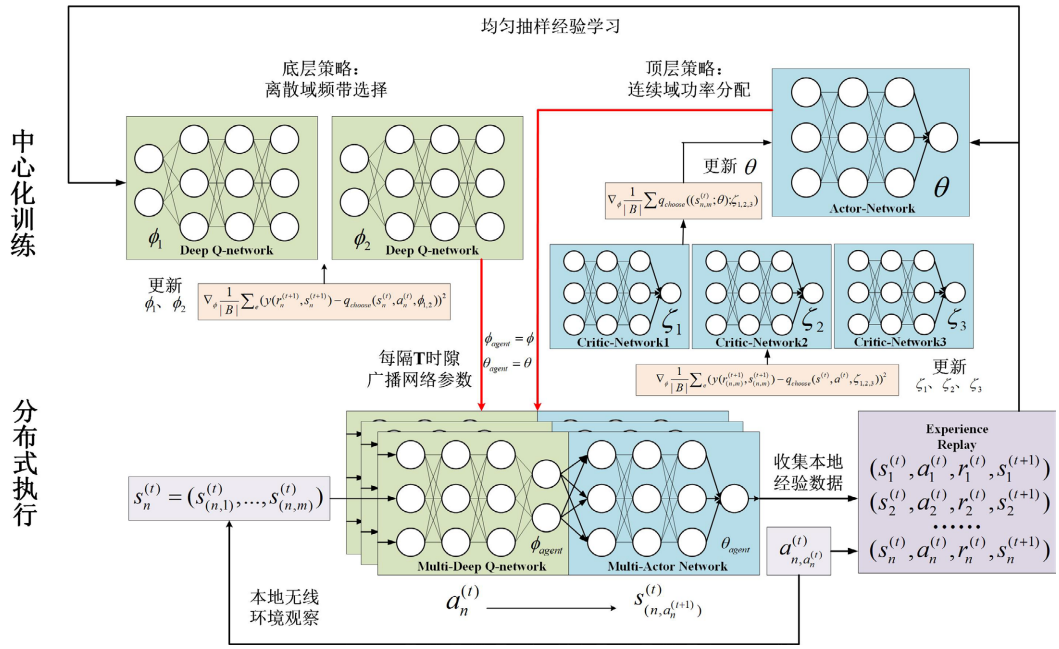


Figure 4. Algorithm structure diagram
图 4. 算法结构图

$$\nabla_{\phi}\frac{1}{|B|}\sum q_{choose}\left(\left(s_{n,m}^{(t)};\theta\right);\zeta_{1,2,3}\right) \tag{19}$$

上升更新参数为 θ 的 Actor 网络学习给定状态 $s_{n,a_n^{(t)}}^{(t)}$ 下输出动作 $a_{n,a_n^{(t)}}^{(t)}$ 的确定性策略 μ。

在每个训练时隙的结尾，中心化训练算法需将学得的网络参数 ϕ、θ 向部署在蜂窝网络小区基站处的智能体广播更新 ϕ_{agent}、θ_{agent}。因此，对于具有相同网络参数的分布式执行智能体而言，对本地通信环境下的不同观察，都可在中心化执行算法处进行学习，以使整个系统受益。分布式智能体根据网络参数 ϕ_{agent} 更新子频带选择动作 $a_n^{(t+1)}$，根据网络参数 θ_{agent} 更新功率控制动作 $a_{n,a_n^{(t+1)}}^{(t+1)}$。

4. 实验结果

4.1. 参数设置

算法的超参数设置见表 1。

本文使用 Hass 信道[19]对蜂窝网络下的通信环境进行模拟。

每个设备的最大移动速度为 2.5 m/s，且每个设备每秒在 $[-0.5,0.5]$m/s 间随机改变速度，并在 $[-0.175,0.175]$rad/s 间随机改变方向。

Table 1. Parameter configuration
表 1. 参数配置

参数	数值
小区半径 R	400 m
载波频率 f_c	2 GHz
相干距离 d_{cor}	10 m

Continued

最大功率电平 P_{max}	38 dBm
最大输出功率 P_{max_out}	37.24 dBm
最小输出功率 P_{min_out}	0.76 dBm
固定时隙持续时间 T_{max}	20 ms
对数正态阴影标准差 σ^S	10 dB
白噪声功率谱密度 σ^2	−114 dBm

设备在 5000 个训练时隙中的移动轨迹如图 5 所示。

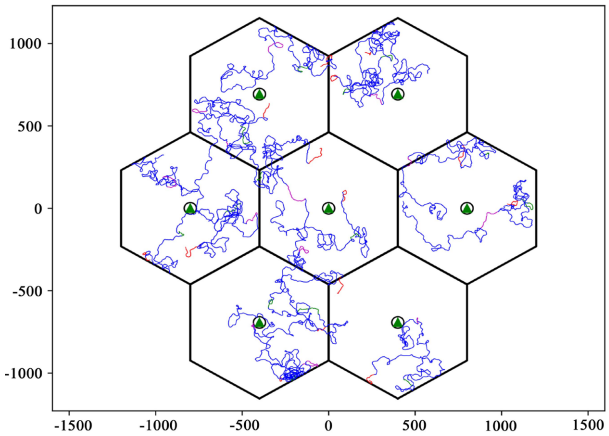


Figure 5. Mobile network deployment example
图 5. 移动网络部署样例

4.2. 移动场景下的控制效果

设 M 为可选用子频带数量，K 为蜂窝小区单元数量，N 为蜂窝网络中移动设备数。训练共计 20,000 个时隙，每 5000 个时隙设为一集，在每集开始时初始化网络部署、学习率等参数。在相同的网络部署环境下，将所提算法(Proposed)与未改进网络结构的分层策略算法[8] (ER)、单网络联合控制算法[6] (Joint)、传统迭代式分数规划算法[3] (FP)、考虑通信延迟的分数规划算法(Delayed FP)进行对比研究，以分析所提算法对函数逼近误差的处理后对算法性能的提升。

图 6 展示了所提算法与文献方法在不同部署方案下的性能对比，并以平均训练奖励变化曲线评价 Proposed 算法与 ER 算法的收敛速度。此外，如表 2 所示，本文取最后 200 个训练时隙的链路平均频谱效率作为算法性能的评价指标。

如图 6(a)、图 6(b)所示，当 $M = 6$ 时，随训练的进行，Proposed 算法与 ER 算法在维持相近收敛速度的同时，其频谱效率远高于其它算法。基于迭代式的算法(FP, Delayed FP)无法通过智能体与环境的交互更新策略，使得其链路平均频谱效率变化幅度不大，无法获得进一步的性能提升。而 Joint 算法收敛速度过于缓慢，且在最后 200 个时隙的性能上只取得了高于迭代式算法的表现。

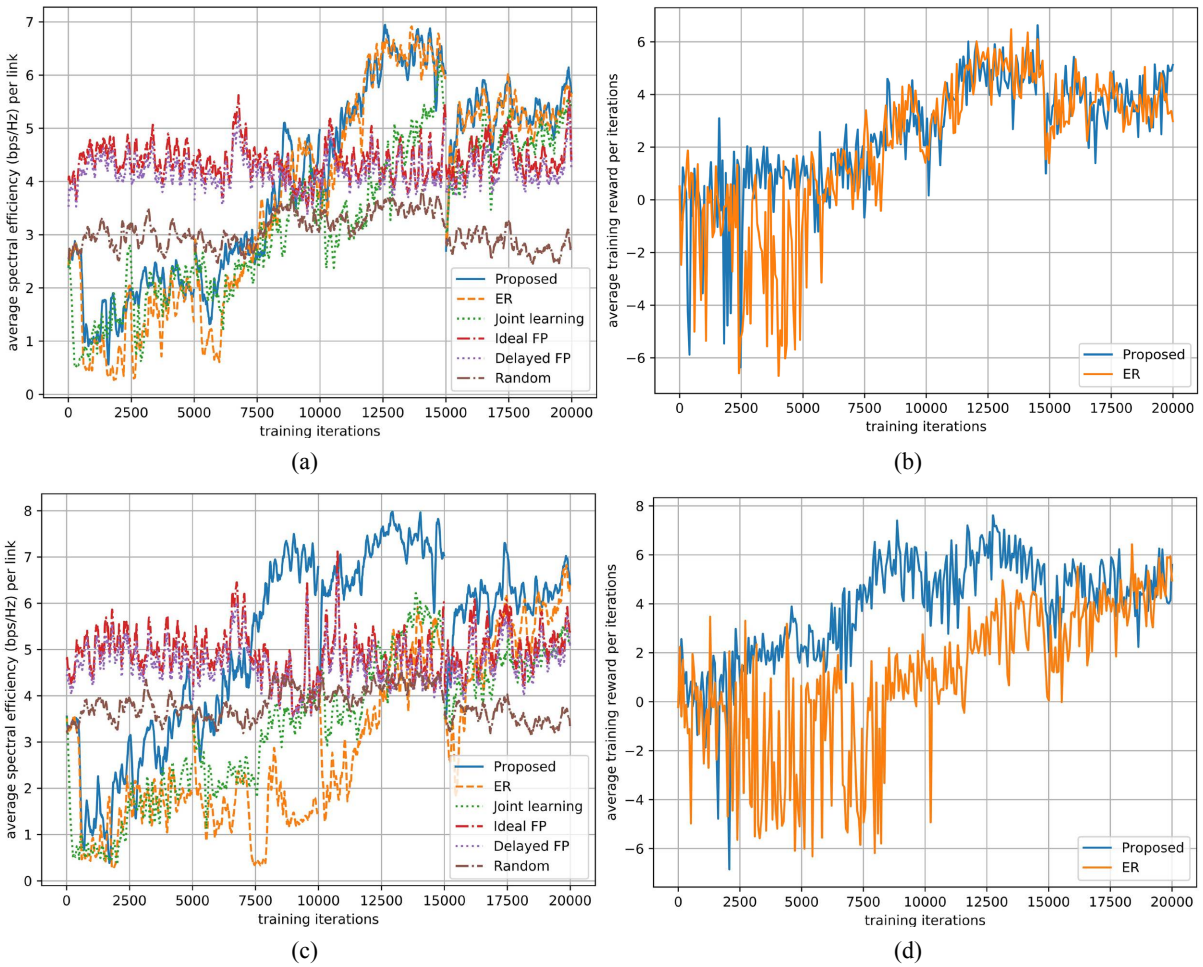
如图 6(c)、图 6(d)所示，当 $M = 8$ 时，随训练的进行，更高的频带数量意味着智能体对本地环境的观测维度上升，以 DQN 与 DDPG 为网络结构的 ER 算法，由于对真实价值估计不准确，此误差随训练的进行逐步累积，使算法收敛至次优策略而降低了算法的收敛速度与最终的平均频谱效率。在此场景下，Propose 算法相比 ER 算法拥有更快的收敛速度，且在整个训练过程中，拥有着最高的链路平均频谱效率。

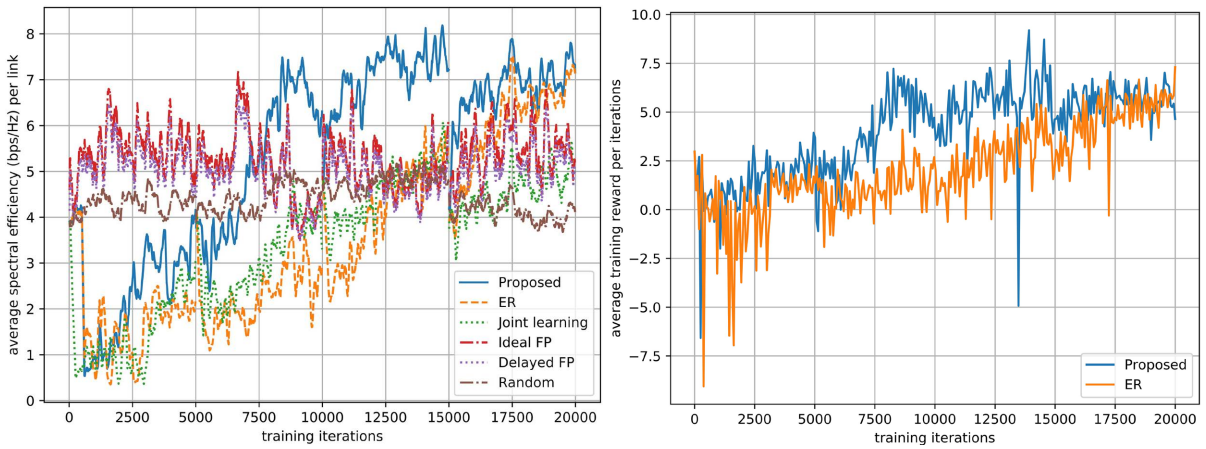
同样的结论在更高频带数场景下仍可观测到。如图 6(e)、图 6(f)所示，除 Proposed 算法外，尽管 ER 算法在第四集上的性能要高于其他算法，但过慢的收敛速度使得该算法已不易部署在实际的蜂窝通信场景下。而本文所提 Proposed 算法在高频带场景下，通过对动作价值更准确的估计，使算法在保证较快收敛速度的同时拥有最高的链路平均频谱效率。这印证了本文在 $M=8$ 场景下的结论。

4.3. 测试集表现

对所有算法测试其经训练的策略在不同频带数量情形下的性能，各次实验采用随机生成的部署方案。表 3 为其测试数据在最后 200 个时隙上的链路平均频谱效率表现。

在 $M \le 4$ 时，以单一网络生成控制策略的 Joint 算法以更多输出层为代价，得到了较高的链路平均频谱效率。而随频带数量的增长，动作空间维度增长迅速，单一网络架构无法处理高维动作空间下智能体所观测的信息，致使频谱效率提升有限。而基于分层策略的 ER 算法，通过将离散策略与连续策略分为两个网络联合学习，使得其在应对高频带场景下的表现要优于 Joint 算法，在 $M \ge 2$ 下均能取得除 Proposed 算法外最高的平均频谱效率。然而，基于 DQN 与 DDPG 为内核的 ER 算法，缺乏对动作值的准确估计，其频谱效率仍拥有进一步的提升空间。而本文所提 Proposed 算法，通过结合 DDQN 与三元组批评机制的 TD3 算法，在高频带场景所带来的高维动作空间下，保证收敛速度的同时，拥有最高的链路平均频谱效率。测试数据表明，预训练策略在随机生成的部署方案中仍然可用，且本文所提算法比参考算法拥有更高的算法性能。





(e) (f)

Figure 6. The control effect of the proposed algorithm in different frequency band scenarios. (a) Comparison of average spectral efficiency of various algorithms with $M = 6$, $(K, N) = (5, 20)$; (b) Comparison of changes in average training rewards for $M = 6$, $(K, N) = (5, 20)$; (c) Comparison of average spectral efficiency of various algorithms with $M = 8$, $(K, N) = (5, 20)$; (d) Comparison of changes in average training rewards for $M = 8$, $(K, N) = (5, 20)$; (e) Comparison of average spectral efficiency of various algorithms with $M = 10$, $(K, N) = (5, 20)$; (f) Comparison of changes in average training rewards for $M = 10$, $(K, N) = (5, 20)$

图 6. 所提算法在不同频带数场景下的控制效果。(a) $M = 6$, $(K, N) = (5, 20)$各算法平均频谱效率对比；(b) $M = 6$, $(K, N) = (5, 20)$平均训练奖励变化对比；(c) $M = 8$, $(K, N) = (5, 20)$各算法平均频谱效率对比；(d) $M = 8$, $(K, N) = (5, 20)$平均训练奖励变化对比；(e) $M = 10$, $(K, N) = (5, 20)$各算法平均频谱效率对比；(f) $M = 10$, $(K, N) = (5, 20)$平均训练奖励变化对比

Table 2. Comparison of algorithm training performance
表 2. 算法训练性能对比

(K, N)	M	链路平均频谱效率(bps/Hz)						强化学习方法输出层大小		平均迭代次数
		强化学习方法			其他学习方案					
		Proposed	ER	Joint	Ideal FP	Delayed FP	Random	Proposed	Joint	FP
(5, 20)	1	1.73	1.72	1.85	1.85	1.85	0.43	1 + 1	10	57.67
	2	2.85	2.75	2.74	2.67	2.59	0.89	2 + 1	20	97.46
	4	4.69	4.53	4.50	3.77	3.65	1.87	4 + 1	40	114.52
	6	5.81	5.59	5.44	4.92	4.72	2.88	6 + 1	60	126.79
	8	6.67	6.49	5.15	5.59	5.34	3.46	8 + 1	80	134.29
	10	7.31	7.04	5.08	5.62	5.41	4.18	10 + 1	100	141.81

Table 3. Comparison of algorithm test performance
表 3. 算法测试性能对比

(K, N)	M	链路平均频谱效率(bps/Hz)						强化学习方法输出层大小		平均迭代次数
		强化学习方法			其他学习方案					
		Proposed	ER	Joint	Ideal FP	Delayed FP	Random	Proposed	Joint	FP
(5, 20)	1	1.59	1.58	1.60	1.56	1.46	0.41	1 + 1	10	70.30
	2	2.80	2.79	2.79	2.68	2.53	0.99	2 + 1	20	105.73
	4	4.59	4.56	4.49	3.71	3.54	2.11	4 + 1	40	121.56

Continued

	6	5.92	5.86	5.52	4.47	4.25	3.04	6 + 1	60	135.08
(5, 20)	8	6.89	6.64	5.64	5.06	4.83	3.81	8 + 1	80	142.30
	10	7.05	6.97	5.71	5.44	5.02	4.46	10 + 1	100	156.86

5. 结论

针对蜂窝网络中的功率控制问题，提出使用一种联合控制策略，使用 DDQN 处理离散子频带选取策略，并结合三元组批评机制改进 TD3 算法学习连续功率控制策略。通过在网络结构上增添 Critic 网络、延迟更新 Actor 网络、添加噪声使得算法对动作价值的估计更加精准，提升算法在各频带场景下的收敛速度与算法性能。实验结果表明，本文所提算法在各种频带数量场景下均拥有较快的收敛速度与较高的链路平均频谱效率。未来，我们正在研究更易于调整的数据抽样与利用策略，并提升算法在多智能体部署下的稳定性。

基金项目

华为技术有限公司合作项目(YBN2019115054)资助。

参考文献

[1] Luo, Z.-Q. and Zhang, S. (2008) Dynamic Spectrum Management: Complexity and Duality. *IEEE Journal of Selected Topics in Signal Processing*, **2**, 57-73. https://doi.org/10.1109/JSTSP.2007.914876

[2] Tan, J., Zhang, L. and Liang, Y.-C. (2019) Deep Reinforcement Learning for Channel Selection and Power Control in D2D Networks. 2019 *IEEE Global Communications Conference* (*GLOBECOM*), Waikoloa, 9-13 December 2019, 1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9014074

[3] Shen, K. and Yu, W. (2018) Fractional Programming for Communication Systems—Part I: Power Control and Beamforming. *IEEE Transactions on Signal Processing*, **66**, 2616-2630. https://doi.org/10.1109/TSP.2018.2812733

[4] Sun, H., Chen, X., Shi, Q., *et al*. (2018) Learning to Optimize: Training Deep Neural Networks for Interference Management. *IEEE Transactions on Signal Processing*, **66**, 5438-5453. https://doi.org/10.1109/TSP.2018.2866382

[5] Tan, J., Liang, Y.-C., Zhang, L. and Feng, G. (2020) Deep Reinforcement Learning for Joint Channel Selection and Power Control in D2D Networks. *IEEE Transactions on Wireless Communications*, **20**, 1363-1378. https://doi.org/10.1109/TWC.2020.3032991

[6] Nasir, Y.S. and Guo, D. (2019) Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks. *IEEE Journal on Selected Areas in Communications*, **37**, 2239-2250. https://doi.org/10.1109/JSAC.2019.2933973

[7] Meng, F., Chen, P., Wu, L. and Cheng, J. (2020) Power Allocation in Multi-User Cellular Networks: Deep Reinforcement Learning Approaches. *IEEE Transactions on Wireless Communications*, **19**, 6255-6267. https://doi.org/10.1109/TWC.2020.3001736

[8] Nasir, Y.S. and Guo, D. (2021) Deep Reinforcement Learning for Joint Spectrum and Power Allocation in Cellular Networks. 2021 *IEEE Globecom Workshops* (*GC Wkshps*), Madrid, 7-11 December 2021, 1-6. https://doi.org/10.1109/GCWkshps52748.2021.9681985

[9] Van Hasselt, H., Guez, A. and Silver, D. (2016) Deep Reinforcement Learning with Q-Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**, 2094-2100. https://doi.org/10.1609/aaai.v30i1.10295

[10] Fujimoto, S., van Hoof, H. and Meger, D. (2018) Addressing Function Approximation Error in Actor-Critic Methods. *Proceedings of the* 35*th International Conference on Machine Learning*, Stockholm, 10-15 July 2018.

[11] Wu, D., Dong, X., Shen, J. and Hoi, S.C.H. (2020) Reducing Estimation Bias via Triplet-Average Deep Deterministic Policy Gradient. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 4933-4945. https://doi.org/10.1109/TNNLS.2019.2959129

[12] Nguyen, T.T., Nguyen, N.D. and Nahavandi, S. (2020) Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*, **50**, 3826-3839. https://doi.org/10.1109/TCYB.2020.2977374

[13] Ren, J., He, Y., Wen, D., *et al*. (2020) Scheduling for Cellular Federated Edge Learning with Importance and Channel Awareness. *IEEE Transactions on Wireless Communications*, **19**, 7690-7703. https://doi.org/10.1109/TWC.2020.3015671

[14] Liang, L., Peng, H., Li, G.Y. and Shen, X. (2017) Vehicular Communications: A Physical Layer Perspective. *IEEE Transactions on Vehicular Technology*, **66**, 10647-10659. https://doi.org/10.1109/TVT.2017.2750903

[15] 陈晓玉, 周佳玲. 分布式强化学习在经济调度问题中的应用[J]. 控制工程, 2022, 29(3): 480-485.

[16] Duan, J., Guan, Y., Li, S.E., Ren, Y., Sun, Q. and Cheng, B. (2021) Distributional Soft Actor-Critic: Off-Policy Reinforcement Learning for Addressing Value Estimation Errors. *IEEE Transactions on Neural Networks and Learning Systems*, **33**, 6584-6598. https://doi.org/10.1109/TNNLS.2021.3082568

[17] 何斌, 刘全, 张琳琳, 等. 一种加速时间差分算法收敛的方法[J]. 自动化学报, 2021, 47(7): 1679-1688.

[18] Zhao, Y., Niemegeers, I.G. and De Groot, S.M.H. (2021) Dynamic Power Allocation for Cell-Free Massive MIMO: Deep Reinforcement Learning Methods. *IEEE Access*, **9**, 102953-102965. https://doi.org/10.1109/ACCESS.2021.3097243

[19] Nasir, Y.S. and Guo, D. (2020) Deep Actor-Critic Learning for Distributed Power Control in Wireless Mobile Networks. 2020 54*th Asilomar Conference on Signals*, *Systems*, *and Computers*, Pacific Grove, 1-5 November 2020, 398-402. https://doi.org/10.1109/IEEECONF51394.2020.9443301