无蜂窝毫米波大规模MIMO系统基于深度强化学习的节能睡眠策略

何 云[1,2],申 敏[1,2],王 蕊[1,2],张 梦[1]
(1. 重庆邮电大学通信与信息工程学院,重庆 400065;2. 重庆邮电大学通信核心芯片、协议及系统应用团队,重庆 400065)

摘 要: 为了提升无蜂窝毫米波大规模MIMO(Cell-Free millimeter-Wave massive MIMO,CF mmWave mMIMO)系统总能量效率,本文研究时变信道环境中接入点(Access Point,AP)睡眠节能机制.将AP开关切换(AP Switch ON-OFF,ASO)策略看作一个马尔可夫决策过程,使用深度强化学习(Deep Reinforcement Learning,DRL)工具解决AP开关问题.引入干扰感知技术和局部敏感哈希检索方法减少代理与复杂环境的交互以及样本偏差,构造了一个新的效用函数,在严格用户服务质量(Quality of Service,QoS)约束下更好地权衡总能效和可达速率性能.通过对效用函数离散化分级处理,将状态空间映射为更小的分级状态空间,以加快决斗深度Q网络(Dueling Deep Q-Network,Dueling DQN)的收敛速度.仿真结果证明了该方案的稳定性、收敛性和严格QoS约束下的总能效性能优势.

关键词: 无蜂窝;毫米波;深度强化学习;AP开关切换;能效
基金项目: 国家科技重大专项基金(No.2018ZX03001026-002)
中图分类号: TN929.5 **文献标识码:** A **文章编号:** 0372-2112(2023)10-2831-13
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220247

Energy-Efficient Sleep-Mode Based on Deep Reinforcement Learning for Cell-Free mmWave Massive MIMO Systems

HE Yun[1,2], SHEN Min[1,2], WANG Rui[1,2], ZHANG Meng[1]
(1. *School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*;
2. *Innovation Team of Communication Core Chip, Protocols and System Application, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

Abstract: To improve the global energy-efficiency (GEE) performance in cell-free millimeter-wave massive MIMO (CF mmWave mMIMO) systems, the access points (APs) sleep-mode techniques in dynamic time-varying channels are investigated. The AP switch ON-OFF (ASO) strategy is formulated as a Markov decision process. Thus, a deep reinforcement learning (DRL) model can be used to solve the AP activation problem. The interference-aware method and the locality-sensitive hashing method are introduced to reduce sample bias and interaction between agents and complex environments. A novel cost function is constructed to achieve a better balance between GEE and achievable rate under the strict quality of service (QoS) constraints. In order to accelerate the convergence of the dueling deep Q-Network (DQN), the state space is mapped to the smaller hierarchical state space by discretizing the cost function. Simulation results have demonstrated the performance advantage of the convergence of deep reinforcement learning and GEE under the strict QoS constraint.

Key words: cell-free; millimeter-wave; deep reinforcement learning; access point switch on-off; energy-efficiency
Foundation Item(s): National Science and Technology Major Project of China (No.2018ZX03001026-002)

收稿日期:2021-10-08;修回日期:2023-09-01;责任编辑:王天慧

1 引言

无蜂窝毫米波大规模MIMO(Cell-Free millimeter-Wave massive MIMO,CF mmWave mMIMO)系统中接入点(Access Point,AP)的密集部署、大规模天线阵列以及前传链路的巨大功率消耗将显著增加网络总能耗,能效优化问题对无蜂窝毫米波大规模MIMO系统的可持续发展具有重要意义[1].在低流量负载情况下,所有接入点都处于开启状态将导致资源的低效利用和能源浪费.AP开关切换(AP Switch ON-OFF,ASO)策略作为众多绿色策略中一种基于网络布局的有效节能策略[2~4],旨在提供能效最大化的AP开关方案,然而,这是一个NP难问题[3],需要评估AP的所有可能组合.一般来说,解决该问题的传统方法有启发式方法和基于模型的方法[5].

启发式方法可以很好地解决静态分布式环境中的资源分配问题,但它们对系统参数很敏感.如果关键变量或约束(如新增AP或用户)发生变化,则需要重新配置参数以生成更可行的解决方案.此外,传统启发式方法不能充分利用网络实时信息,无法基于用户和AP的分布信息、流量负荷等信息来优化AP开关状态,无法对动态变化环境做出响应[6].不同于线性和时不变移动系统,现实通信网的关键问题需要考虑用户移动性和信号传播随机性,开发出适应复杂时变信道环境不可预测衰落和强阴影效应的解决方案.

基于模型驱动的机器学习算法能够从数据中学习有意义的信息,通过最小化人工干预获得最佳性能.基于模型的方法易于分析,但训练数据的采集不仅给通信协议带来极大负担[7],同时动态变化的环境也使训练数据不易获取[8],因此,无模型和数据驱动的方法更有前途.强化学习技术源于大数据分析,作为一种数据驱动的资源管理方法,可在未知动态变化网络条件(如可变信道状态信息和用户服务质量)下为每个状态交互做出最佳决策.

传统的强化学习算法使用Q表来存储每个状态-动作对的奖励值,但是高维Q表不仅会使学习过程难以收敛,而且手动提取特征工程极其耗时,且特征不完整.这些缺陷限制了强化学习技术在无线通信系统中的实际应用.为了避免强化学习算法的不稳定性,Mnih等人[9]中首次提出基于深度Q网络(Deep Q-Network,DQN)的深度强化学习(Deep Reinforcement Learning,DRL)方法.深度Q学习作为经典Q学习算法的一种变体,结合了强化学习技术和深度神经网络(Deep Neural Network,DNN)技术,允许DRL代理基于原始数据的分层组合,使用高维神经网络数据的强大表示能力,从而避免状态-动作的手动输入.然而,在无蜂窝毫米波大规模MIMO系统中使用深度强化学习技术解决能效优化问题需要面临以下挑战.

首先,虽然ASO策略能有效提高无蜂窝毫米波大规模MIMO系统的总能效[6],但大部分研究忽略了用户服务质量(Quality of Service,QoS)约束,而系统总能效最大化只有在满足QoS要求时才有意义,否则会降低用户体验[10].QoS要求是以增加系统功耗为代价的.为了解决QoS约束下的节能资源分配问题,人们引入深度强化学习工具并提出了一些新的节能策略,如文献[11,12]基于强化学习模型,提出了表征可达速率和能效之间权衡的效用函数来解决蜂窝网络的基站激活问题.但是以上研究很难识别时变信道环境中可达速率和总能效的权重边界,且效用函数往往具有松散的QoS约束,而非严格的QoS约束.因此无蜂窝毫米波大规模MIMO系统的AP开关策略需要关注严格QoS约束下总能效和可达速率性能的权重设计.其次,无蜂窝毫米波大规模MIMO系统前传链路上大量高维信道状态信息(Channel State Information,CSI)的频繁信令交换会对传输实时性造成巨大压力,且AP开关状态空间大小或者动作空间大小随着AP数的增加呈指数增长[12],这些都会降低深度强化学习网络的学习速度.最后,对时变信道的采样会使数据分布随着学习新行为而发生变化,这对于假设固定分布的深度学习模型来说是有问题的.无线网络不同时刻收集的样本会使相同状态转移的即时奖励不唯一,即学习过程中产生样本偏差,从而使强化学习过程不易收敛.

使用深度强化学习工具解决QoS约束下AP开关选择问题时,当某用户的QoS不满足需求时,继续关闭服务于该用户的AP不能保障该用户的服务体验,使关闭特定AP的动作没有意义.而决斗深度强化学习方法在深度强化学习框架基础上增加了动作重要性评估,能有效提升上述场景中强化学习性能.决斗深度强化学习方法在许多应用中已经证明了其有效性[13].为了解决强化学习中的样本偏差问题,将状态-动作对等样本信息进行缓存处理,从缓存空间中找到有效样本代替偏差样本是一种有效策略,该策略应用于实时通信系统时面临的一个关键问题是要提高检索效率.近邻检索方法能以较低成本实现在线搜索,基于近邻检索的局部敏感哈希(Locality Sensitive Hashing,LSH)方法以其次线性搜索时间和可控的成功概率而受到广泛关注[14].基于欧几里德LSH函数的方法是一种典型的LSH方法,但是其显著缺点是会消耗大量内存.为了解决该问题,面向数据的投影向量方法能提升哈希函数的性能[15].

基于以上分析,本文利用无线通信领域知识和支撑实时通信的深度强化学习理论,采用决斗深度强化学习模型设计了一种AP开关策略,以实现严格QoS约

束下总能效和可达速率的性能权衡. 首先,本文基于哈希检索技术提出了一种缓存策略以降低样本误差并减少不必要的信令交互. 然后,本文基于干扰感知技术提出了一种低复杂度效用函数计算方法,该算法无需使用高维CSI信息,推导的最优权重的闭解形式能在严格QoS约束下实现总能效和可达速率的性能权衡. 最后,本文提出了一种状态空间分级算法和单步动作转移建模策略,二者分别通过减小状态空间大小和动作空间大小来提高强化学习算法的收敛速度.

2　系统模型

在无蜂窝毫米波大规模MIMO系统中,大量AP协同为用户服务,AP通过前传链路连接到中央处理单元(Central Processing Unit, CPU). 假设系统配置有 M 个AP和 K 个用户($M \gg K$). 系统工作在时分双工(Time-Division Duplex, TDD)模式,根据信道互易性通过上行训练获取CSI信息. 无蜂窝毫米波大规模MIMO系统采用以用户为中心的分簇方法来降低前传链路的功耗及系统复杂度,允许每个用户由特定AP集合提供最好服务,以用户为中心的无蜂窝毫米波大规模MIMO系统结构如图1所示.

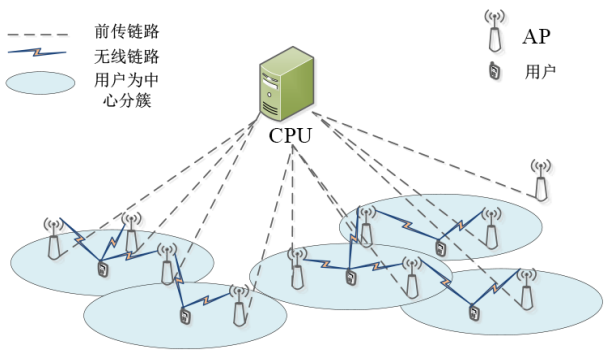


图1　无蜂窝毫米波大规模MIMO系统结构

假设为第 k 个用户提供服务的AP集合为 $\mathcal{M}_k$,集合 $\mathcal{M}_k$ 包含的最大AP数为 $N_{\mathrm{AP,max}}$. 系统采用基于簇的大规模MIMO信道模型[16],第 m 个AP和第 k 个用户之间的信道 $\boldsymbol{H}_{m,k} \in \mathbb{C}^{N_t \times N_{r,k}}$ 为

$$\boldsymbol{H}_{m,k} = \sqrt{\frac{N_t N_{r,k}}{N_{\mathrm{cl}} N_{\mathrm{ray}}}} \sum_{i=1}^{N_{\mathrm{cl}}} \sum_{l=1}^{N_{\mathrm{ray}}} \alpha_{i,l} \sqrt{L\left(d_{i,l}\right)} \times \boldsymbol{a}_{\mathrm{AP}}\left(\theta_{i,l,m,k}^{\mathrm{AP}}\right) \boldsymbol{a}_{\mathrm{UE}}^{\mathrm{H}}\left(\theta_{i,l,m,k}^{\mathrm{UE}}\right) + \boldsymbol{H}_{\mathrm{LoS}} \tag{1}$$

$$\boldsymbol{H}_{\mathrm{LoS}} = I(d) \sqrt{N_t N_{r,k}}\, \mathrm{e}^{\mathrm{j}\eta} \sqrt{L(d)}\, \boldsymbol{a}_{\mathrm{AP}}\left(\theta_{\mathrm{LoS}}^{\mathrm{AP}}\right) \boldsymbol{a}_{\mathrm{UE}}^{\mathrm{H}}\left(\theta_{\mathrm{LoS}}^{\mathrm{UE}}\right) \tag{2}$$

其中 N_{cl} 和 N_{ray} 分别为散射簇数和每个簇的传播路径数,$\boldsymbol{a}_{\mathrm{AP}}(\theta)$ 和 $\boldsymbol{a}_{\mathrm{UE}}(\theta)$ 分别为AP和用户的阵列响应向量. $\theta_{i,l,m,k}^{\mathrm{UE}}$ 和 $\theta_{i,l,m,k}^{\mathrm{AP}}$ 分别为第 k 个用户和第 m 个AP在路径 (i,l) 上的到达角和离开角,$a_{i,l} \sim \mathrm{CN}(0,1)$ 是在路径 (i,l) 上的复数增益. $\boldsymbol{H}_{\mathrm{LoS}}$ 是直视径信道(Line-of-Sight, LoS),$\eta \sim \mathcal{U}(0, 2\pi)$, $I(d)$ 是0-1随机分布变量,用于指示在AP和用户间是否存在LoS链路. $d_{i,l}$ 为AP和用户在路径 (i,l) 上的距离,$L(d)$ 为AP和用户之间的路径损耗.

$$L(d) = -20 \log_{10}\left(\frac{4\pi f_0}{c}\right) - 10n \log_{10}(d) - X_\sigma \tag{3}$$

其中,n 是路径衰落因子,f_0 是载波频率,c 是光速. X_σ 是阴影衰落项,其均值为零,方差为 σ^2.

时刻 t 的AP开关状态 $s^{(t)}$ 定义为

$$s^{(t)} = [o_1, o_2, \cdots, o_M] \tag{4}$$

其中,$o_m = 1$ 表示第 m 个AP处于开启状态,$o_m = 0$ 表示第 m 个AP处于关闭状态. 时刻 t 处于开启状态的AP集合定义为 $O = \{m: o_m = 1,\ s^{(t)} = [o_1, o_2, \cdots, o_M]\}$, 以 $s^{(t)}$ 为目标的第 k 个用户下行可达速率 $\mathcal{R}_k^d(s^{(t)})$ 为

$$\begin{aligned} \mathcal{R}_k^d(s^{(t)}) &= B_0 \log_2(1 + \gamma_k(s^{(t)})) \\ &= B_0 \log_2 \left| \boldsymbol{I} + \boldsymbol{R}_k^{-1} \boldsymbol{A}_{k,k} \boldsymbol{A}_{k,k}^{\mathrm{H}} \right| \end{aligned} \tag{5}$$

$$\boldsymbol{R}_k = \sum_{l \neq k} \boldsymbol{A}_{k,l} \boldsymbol{A}_{k,l}^{\mathrm{H}} + \sigma_n^2 \boldsymbol{L}_k^{\mathrm{H}} \boldsymbol{L}_k \tag{6}$$

$$\boldsymbol{A}_{k,l} = \sum_{\substack{m \in O, \\ m \in \mathcal{M}_k}} \sqrt{p_{m,l}}\, \boldsymbol{L}_k^{\mathrm{H}} \boldsymbol{H}_{m,k}^{\mathrm{H}} \boldsymbol{F}_{m,l} \tag{7}$$

其中,$\gamma_k(s^{(t)})$ 为第 k 个用户在状态 $s^{(t)}$ 的信干噪比(Signal to Interference plus Noise Ratio, SINR),σ_n^2 为噪声功率,每个用户的带宽为 B_0. $\boldsymbol{F}_{m,k}$ 为第 m 个AP服务于第 k 个用户的混合预编码器,$\boldsymbol{L}_k$ 为第 k 个用户的合并器,$\boldsymbol{R}_k$ 为第 k 个用户的干扰协方差矩阵加上有效噪声.

系统总能效 $\mathcal{K}^d(s^{(t)})$ 表示为[16]

$$\mathcal{K}^d(s^{(t)}) = \frac{\mathcal{R}^d(s^{(t)})}{P_{\mathrm{T}}(s^{(t)})} = \frac{\sum_{k=1}^{K} \mathcal{R}_k^d(s^{(t)})}{P_{\mathrm{T}}(s^{(t)})} \tag{8}$$

$$\begin{aligned} P_{\mathrm{T}}(s^{(t)}) = &\sum_{k=1}^{K} \left[\sum_{\substack{m \in O, \\ m \in \mathcal{M}_k}} \left(\frac{p_{m,k}}{\delta} \right) \right] + \sum_{m \in O} P_{c,m} \\ &+ \sum_{k=1}^{K} \mathcal{R}_k^d(s^{(t)}) P_{\mathrm{fh},m} + \sum_{m \in O} P_{\mathrm{fh},0,m} + \sum_{m \notin O} P_{\mathrm{fh},m,\mathrm{sleep}} \end{aligned} \tag{9}$$

其中 $P_{\mathrm{T}}(s^{(t)})$ 为系统总功耗,式右侧的第一项是功率放大器产生的功耗,用户的传输功率为 $p_{m,k}$,第二项为本地振荡器(Local Oscillator, LO)、移相器和射频链路消耗的硬件功耗[17]. 其他项为前传链路产生的功耗,处于开启状态AP的前传链路固定功耗为 $P_{\mathrm{fh},0,m}$,处于关闭状态AP的前传链路固定功耗为 $P_{\mathrm{fh},m,\mathrm{sleep}}$,参数定义及描述参见表1.

3　基于深度强化学习的AP开关算法

3.1　问题建模

以 $s^{(t)}$ 为优化目标的总能效优化问题P1为

$$\begin{aligned}&\mathrm{P1}:\max_{s^{(t)}}\quad \mathcal{K}^d\left(s^{(t)}\right)\\&\text{s.t. C1}:\sum_{k=1}^{K}p_{m,k}\leqslant P_{\max}\\&\quad\mathrm{C2}:p_{m,k}\geqslant 0,\forall m=1,2,\cdots,M\\&\quad\mathrm{C3}:\mathcal{R}_k^d(s^{(t)})\geqslant B_0 r_{\min}\end{aligned}\tag{10}$$

最大功率约束C1中AP最大功率为$P_{\max}$,非负功率约束为C2. QoS约束C3中用户最小频谱效率为$r_{\min}$. 由于总能效的提升是以降低用户可达速率为代价的,本文的AP开关策略需要实现总能效和可达速率性能权衡,定义性能权衡效用函数$\varsigma\left(\mathcal{R}_k^d,\mathcal{K}^d\right)$为

$$\varsigma\left(\mathcal{R}_k^d,\mathcal{K}^d\right)=\mu\xi\left(\mathcal{R}_k^d\right)+(1-\mu)\psi\left(\mathcal{K}^d\right)\tag{11}$$

$$\xi\left(\mathcal{R}_k^d\right)=\begin{cases}-1, & \mathcal{R}_k^d<R_{\min}\\ \dfrac{\mathcal{R}_k^d-R_{\min}}{R_{\max}-R_{\min}}, & \mathcal{R}_k^d\geqslant R_{\min}\end{cases}\tag{12}$$

$$\psi\left(\mathcal{K}^d\right)=\frac{1}{1+\mathrm{e}^{-\mathcal{K}^d/\omega}}\tag{13}$$

其中,$R_{\max}$为用户最大可达速率,ω为归一化参数. $\xi\left(\mathcal{R}_k^d\right)$和$\psi\left(\mathcal{K}^d\right)$分别表示可达速率满意度函数和总能效满意度函数. 当用户可达速率不满足QoS需求时,以负值惩罚$\xi\left(\mathcal{R}_k^d\right)$函数. μ为加权系数,取值在0和1之间. QoS约束下总能效优化问题P1转化为以μ和$s^{(t)}$为目标的效用函数优化问题P2:

$$\begin{aligned}&\mathrm{P2}:\max_{s^{(t)},\mu}\quad \varsigma\left(\mathcal{R}_k^d\left(s^{(t)}\right),\mathcal{K}^d\left(s^{(t)}\right)\right)\\&\text{s.t. C1, C2}\end{aligned}\tag{14}$$

大规模MIMO天线阵列结构使高维CSI信息的计算涉及到复杂的矩阵运算,为了降低高维CSI信息交互对前传链路的压力,本文引入干扰感知技术,通过测量干扰信号功率和有用信号功率代替CSI信息. 假设时刻$t-\Delta T$与时刻t之间AP状态保持不变,CPU在时间间隔ΔT内测量的所有用户干扰功率和有用信号功率构成大小M_t的样本集合$\boldsymbol{I}_k=\left\{I_{k,n\Delta t}\right\}$,$\boldsymbol{g}_k=\left\{g_{k,n\Delta t}\right\}$,$\Delta t=\Delta T/M_t$,$n=1,2,\cdots,M_t$,则平均干扰功率为$\bar{I}_k(s^{(t)})=\sum_{n=1}^{M}I_{k,n\Delta t}/M_t$,平均有用信号功率为$\bar{g}_k(s^{(t)})=\sum_{n=1}^{M}g_{k,n\Delta t}/M_t$,平均信干噪比为

$$\bar{\gamma}_k(s^{(t)})=\frac{\bar{g}_k\left(s^{(t)}\right)}{\bar{I}_k\left(s^{(t)}\right)+\sigma^2}\tag{15}$$

基于干扰感知技术,平均效用函数优化问题P3为

$$\begin{aligned}&\mathrm{P3}:\max_{s^{(t)},\mu}\ \varsigma^{(t)}\\&\text{s.t. C1, C2}\end{aligned}\tag{16}$$

$$\varsigma^{(t)}=\varsigma\left(\bar{\mathcal{R}}_k^d(s^{(t)}),\bar{\mathcal{K}}^d\left(s^{(t)}\right)\right)\tag{17}$$

其中,$\bar{\mathcal{R}}_k^d(s^{(t)})$和$\bar{\mathcal{K}}^d\left(s^{(t)}\right)$分别为平均可达速率和平均总能效. 传统的效用函数难以确定优化问题P3的加权系数[12],本文提出性质1来确定加权系数μ.

性质1 $\mathcal{K}_{\max}$定义为无QoS约束下的最大总能效,$\mathcal{K}_{\min}$定义为最大可达速率对应的总能效. 给定状态$s^{(t)}$,式(18)的近似解μ为

$$\mu=\frac{\psi\left(\mathcal{K}_{\max}\right)-\psi\left(\mathcal{K}_{\min}\right)}{\psi\left(\mathcal{K}_{\max}\right)-\psi\left(\mathcal{K}_{\min}\right)+2}\tag{18}$$

性质1的证明过程请参见附录A.

3.2 算法设计

在优化问题P3中,从大量AP中选择服务于特定用户的AP开关状态是一个NP难问题,即使在时不变环境中获得最优解,其计算复杂度仍然很高. 考虑到通信能力的限制,由于难以获得足够的先验信息,传统的深度学习方法不可行,且无蜂窝毫米波大规模MIMO通信的实时需求也使深度学习方法无法在有限时间内完成训练过程. 实际上,系统中AP开关状态可以通过不断"试错"学习过程的经验累积实现,因此特别适用于采用强化学习框架来解决该问题.

为了解决AP开关问题,传统强化学习算法定义的动作空间大小随AP数增加呈指数增长[12],无蜂窝毫米波大规模MIMO系统中大量AP将导致动作空间非常大,为了减少动作空间大小来降低强化学习方案复杂度,本文使用了一个简单的动态机制,即将AP开关问题建模为迷宫问题模型. 从初始状态开始,状态之间采用单步动作转移策略,代理沿着运动轨迹收集状态转移信息并给出基本判断,沿着该路径转移的动作轨迹为代理的实际动作. 经过一段时间的学习,当转移到最优状态时,奖励函数为零,AP开关状态收敛到一个稳定的最优状态. 该状态转移过程为具有零吸收态的确定性马尔科夫过程,沿着有限步骤S的运动轨迹,马尔科夫的延迟奖励$R=\sum_{t=1}^{S}\gamma^t r^{(t)}$的最大值将收敛到一个稳定值,其中$\gamma$表示长期奖励的折扣因子. 强化学习基础理论请参见附录B.

强化学习所有要素表示为$\mathcal{E}=\{\mathcal{S},\mathcal{A},\mathcal{R},\mathcal{S}^{'}\}$,其中$\mathcal{S},\mathcal{A},\mathcal{R},\mathcal{S}^{'}$分别为状态空间、动作空间、奖励空间和下一个状态空间. 为了解决AP开关问题,本文定义强化学习的状态为AP开关状态$s^{(t)}$,$s^{(t)}$的所有可能集合组成状态空间,状态空间大小为2^M. 强化学习的动作定义为$a^{(t)}$,其取值范围为0~M,动作空间大小为$M+1$. $a^{(t)}=0$表示维持当前状态不变,$a^{(t)}$取其他值表示第$a^{(t)}$个AP采取与之前状态相反的动作. 例如,时刻t第i个AP处于开启状态,则$a^{(t)}=i$表示下一时刻将采取关闭第i个AP的动作. 强化学习算法在每次迭代过程中使用贪婪机制选择一个动作,即在状态$s^{(t)}$以概率ε选择一个随机动作$a^{(t)}$. 一旦选定该动作,将会获得系统的即时奖励和

下一个状态 $s^{(t+1)}$. 为了表征强化学习中状态转移的正向奖励或者负向奖励,本文定义状态转移产生的即时奖励 $r^{(t)}$ 为采取动作 $a^{(t)}$ 后,状态 $s^{(t+1)}$ 和状态 $s^{(t)}$ 之间效用函数的差值,即

$$r^{(t)}=\varsigma^{(t+1)}-\varsigma^{(t)} \tag{19}$$

传统强化学习方法通常采用 $\mathcal{Q}$ 表来存储状态转移信息,中等规划任务环境中 $\mathcal{Q}$ 表易获取,而在更复杂的无蜂窝毫米波大规模MIMO任务环境中,较大AP数导致 $\mathcal{Q}$ 表维度非常大,使代理性能恶化,因此需要新的机制来解决该问题. 基于DQN网络的深度强化学习框架无需提前获取 $\mathcal{Q}$ 表且不需要大量先验知识,相对于其他深度学习框架具有更快的学习速度,从而能降低代理复杂度. 因此,本文基于DQN网络的深度强化学习算法解决无蜂窝毫米波大规模MIMO系统AP开关问题,采用深度神经网络的近似 $\mathcal{Q}$ 函数来提升算法收敛速度.

本文采用经验回放机制将所有经验都存储在内存池 D 中,从 D 中随机选择少量训练样本馈送到深度神经网络中用于训练[13]. 当内存池 D 中样本收集满后,才进行DQN网络训练,该机制通过多次学习以前的经验来提高学习过程的稳定性. 在内存池收集样本的初始阶段,总能效样本集合为 $K=\left[\mathcal{K}^d\left(s^{(1)}\right),\mathcal{K}^d\left(s^{(2)}\right),\cdots,\mathcal{K}^d\left(s^{(|D|)}\right)\right]$,可达速率集合为 $R=\left[\mathcal{R}_k^d(s^{(1)}),\mathcal{R}_k^d(s^{(2)}),\cdots,\mathcal{R}_k^d(s^{(|D|)})\right]$. 在深度神经网络训练阶段,代理实时收集样本信息更新集合 K 和 R. 本文利用集合 K 和 R 中的统计信息确定效用函数的关键参数 $\mathcal{K}_{\max}$ 和 $\mathcal{K}_{\min}$,以计算式的加权系数. $\mathcal{K}_{\max}$ 和 $\mathcal{K}_{\min}$ 的计算过程如下:

$$\mathcal{K}_{\max}=\underset{\mathcal{K}^d\left(s^{(t)}\right)\in\boldsymbol{K}}{\operatorname{argmax}}\ \mathcal{K}^d\left(s^{(t)}\right) \tag{20}$$

$$\mathcal{K}_{\min}=\mathcal{K}^d\left(s^{(t^*)}\right) \tag{21}$$

$$t^*=\underset{\mathcal{R}_k^d(s^{(t)})\in\boldsymbol{R}}{\operatorname{argmax}}\ \mathcal{R}_k^d(s^{(t)}) \tag{22}$$

为了解决超大状态空间导致的深度强化学习算法收敛缓慢的问题,传统强化学习技术通常采用聚类方法将整个状态空间划分为更小的子空间来降低状态空间大小[11],但是这些方法需要提前收集大量样本进行离线训练. 因此,本文提出将效用函数进行离散化分级处理的方式来降低状态空间大小,将原大小为 2^M 的状态空间映射为大小为 P 的分级状态空间. 分级样本集合为 $\tilde{\mathcal{D}}=\left\{\tilde{\mathcal{D}}_1,\tilde{\mathcal{D}}_2,\cdots,\tilde{\mathcal{D}}_P\right\}$,由分级状态 $\tilde{s}_p$ 和分级效用函数值 $\tilde{\varsigma}_p$ 构成,即 $\tilde{\mathcal{D}}_p=\left\{\tilde{s}_p,\tilde{\varsigma}_p\right\}$,分级算法分两步计算 $\tilde{\varsigma}_p$ 和 $\tilde{s}_p$.

首先,将连续效用函数值区间 $\left[\varsigma_{\min},\varsigma_{\max}\right]$ 从小到大离散化为 P 个分级效用函数值 $\tilde{\varsigma}_p$, $p=1,2,\cdots,P$,即

$$\tilde{\varsigma}_p=\varsigma_{\min}+\Delta\varsigma\times p \tag{23}$$

其中 $\tilde{\varsigma}_p=\varsigma_{\min}+\Delta\varsigma\times p$,$\Delta\varsigma=(\varsigma_{\max}-\varsigma_{\min})/P$. $\varsigma_{\min}$ 为样本收集阶段的最小效用函数值,$\varsigma_{\max}$ 为最大效用函数值. 采用遍历搜索法得到与效用函数值 $\varsigma^{(t)}$ 距离最近的分级效用函数值 $\tilde{\varsigma}_p$,即

$$p=\underset{p=1,2,\cdots,P}{\arg\min}\left|\varsigma^{(t)}-\tilde{\varsigma}_p\right| \tag{24}$$

然后,在分级效用函数 $\tilde{\varsigma}_p$ 的邻近状态集合 Q_p 中找到总能效最大的分级状态 $\tilde{s}_p$,即

$$Q_p=\{i:\varsigma^{(t)}-\tilde{\varsigma}_p<\varsigma^{(t)}-\tilde{\varsigma}_{p'},\mathcal{D}_t\in\mathcal{D},p=1,2,\cdots,P,p\neq p'\} \tag{25}$$

$$\tilde{s}_p=s^{(t^\star)} \tag{26}$$

$$t^\star=\underset{t\in\mathbb{Q}_p}{\arg\max}\ \bar{\mathcal{K}}^d\left(s^{(t)}\right) \tag{27}$$

其中 $\mathcal{D}=\left\{\mathcal{D}_1,\mathcal{D}_2,\cdots,\mathcal{D}_T\right\}$ 为时间周期 T 内的样本集合,$\mathcal{D}_t=\left\{s^{(t)},\varsigma^{(t)}\right\}$ 为时刻 t 的样本. 本文并未采用离线训练的方式获得 $\tilde{s}_p$,而是代理在与环境交互过程中与 Q_p 内最优状态进行比较实时更新 $\tilde{s}_p$,从而避免传统的离线分类训练过程非实时带来的不准确性. 分级算法将实时样本 $\mathcal{D}_t=\left\{s^{(t)},\varsigma^{(t)}\right\}$ 映射到分级样本 $\tilde{\mathcal{D}}_p$,采用了性能权衡作为划分准则,该准则能更合理地反映状态空间的性能差异,从而避免依靠单一评价准则带来的弊端. 状态空间分级算法实现流程如算法1所示,代理首先从样本 $\mathcal{D}_t$ 中取出 $s^{(t)}$ 的效用函数 $\varsigma^{(t)}$,然后基于分级算法得到分级样本集合 $\tilde{\mathcal{D}}$,最后利用以上信息在算法1的第7行输出 $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$.

算法1　状态空间分级算法

输入: 样本集合 $\mathcal{D}_t$ 和分级样本集合 $\tilde{\mathcal{D}}$.

输出: $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$ 和分级样本集合 $\tilde{\mathcal{D}}$.

1. 从 $\mathcal{D}_t$ 中得到 $\varsigma^{(t)}$.
2. 根据式得到 $\varsigma^{(t)}$ 对应的等级 p 和 $\tilde{\varsigma}_p$.
3. 根据 $\tilde{\varsigma}_p$ 计算 $r^{(t)}=\tilde{\varsigma}_p-\varsigma^{(t-1)}$.
4. 从 $\tilde{\mathcal{D}}$ 中取出等级 p 的分级样本 $\tilde{\mathcal{D}}_p$,根据式计算 $\tilde{s}_p$ 并更新 $\tilde{\mathcal{D}}_p$.
5. 输出 $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)=\left(\tilde{s}_p,a^{(t)},r^{(t)},s^{(t+1)}\right)$.

3.3　算法实现

本文基于DQN网络的深度强化学习技术设计无蜂窝毫米波大规模MIMO系统AP开关算法,由于深度神经网络使用了非线性函数的近似处理,算法变得不稳定,从而产生误差. 这主要是由于 $\mathcal{Q}$ 值的小幅度更新会引起策略的巨大变化,导致估计 $\mathcal{Q}$ 值和目标 $\mathcal{Q}$ 值之间相关性以及数据分布受到很大影响,因此,本文引入目标 $\hat{\mathcal{Q}}$ 网络来提高算法稳定性[13],该网络频繁且缓慢地随着 $\mathcal{Q}$ 网络参数的变化更新自身参数,并通过最小化目标 $\hat{\mathcal{Q}}$ 值和估计 $\mathcal{Q}$ 值之间的损失函数更新网络参数,

可以消除估计 $\mathcal{Q}$ 值和目标 $\hat{\mathcal{Q}}$ 值之间的相关性，从而获得良好的学习性能.

深度 $\mathcal{Q}$ 网络利用权重 θ 来参数化 $\mathcal{Q}$ 函数，即 $\mathcal{Q}\left(s^{(t)},a^{(t)};\theta\right)$. 假设 θ_i 和 $\bar{\theta}_i$ 分别为 $\mathcal{Q}$ 网络和目标 $\hat{\mathcal{Q}}$ 网络的参数，则估计 $\mathcal{Q}$ 值和目标 $\hat{\mathcal{Q}}$ 值分别表示为 $\mathcal{Q}\left(s^{(t+1)},a^{(t+1)};\theta_i\right)$ 和 $\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right)$. 在初始阶段，代理采用随机参数初始化两个网络参数. 在第 i 次迭代，代理首先基于给定的 $\hat{\mathcal{Q}}$ 值，通过梯度下降法最小化 $\mathcal{Q}$ 网络的损失函数 $L_i(\theta_i)$ 来更新 $\mathcal{Q}$ 网络参数.

$$L_i(\theta_i)=\mathrm{E}\left[\left(r^{(t)}+\gamma\max_{a^{(t+1)}\in\mathcal{A}}\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right)-\mathcal{Q}\left(s^{(t)},a^{(t)};\theta_i\right)\right)^2\right] \tag{28}$$

$$\nabla_{\theta_i}L_i(\theta_i)=\mathrm{E}\left[\left(r^{(t)}+\gamma\max_{a^{(t+1)}\in\mathcal{A}}\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right)\right)\right]-\mathrm{E}\left[\left(\mathcal{Q}\left(s^{(t)},a^{(t)};\theta_i\right)\nabla_{\theta_i}\mathcal{Q}\left(s^{(t)},a^{(t)};\theta_i\right)\right)\right] \tag{29}$$

然后，为了更新目标 $\hat{\mathcal{Q}}$ 网络，代理使 $\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right)$ 逼近目标值 $y_i^{(t)}$，并采用随机梯度下降法最小化损失函数 $L_i^{tar}(\bar{\theta}_i)$ 来更新权重 $\bar{\theta}_{i+1}$.

$$y_i^{(t)}=r^{(t)}+\gamma\max_{a^{(t+1)}}\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right) \tag{30}$$

$$L_i^{tar}(\bar{\theta}_i)=\left(\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right)-y_i^{(t)}\right)^2 \tag{31}$$

$$\bar{\theta}_{i+1}=\bar{\theta}_i+\bar{\nu}^{(t)}\left(y_i^{(t)}-\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right)\right)\nabla_{\bar{\theta}_i}\hat{\mathcal{Q}}\left(s^{(t+1)},a^{(t+1)};\bar{\theta}_i\right) \tag{32}$$

当某用户的QoS不满足要求时，继续关闭为之服务的AP不能保障该用户的服务体验，因此该状态下采取关闭这些AP的动作是没有意义的. 针对该场景，本文使用了一种基于文献[18]的决斗深度强化学习算法来进一步提高深度强化学习算法性能. 该算法使用值函数来衡量系统给定状态的好坏程度，使用优势函数来衡量给定动作的重要性. 当执行某个动作对系统没有益处时，代理不再估计该动作的值. 假设值函数定义为 $\mathcal{V}(s^{(t)};\theta^{(1)})$，动作 $a^{(t)}$ 的优势函数定义为 $\mathcal{G}(s^{(t)},a^{(t)};\theta^{(2)})$. 决斗深度强化学习算法产生两个数据流分别估计值函数 $\mathcal{V}(s^{(t)};\theta^{(1)})$ 和优势函数 $\mathcal{G}(s^{(t)},a^{(t)};\theta^{(2)})$，其中 $\theta^{(1)}$ 和 $\theta^{(2)}$ 为全连接层的参数. 深度神经网络输出层输出两个流之和作为 $\mathcal{Q}$ 值，即

$$\mathcal{Q}(s^{(t)},a^{(t)};\theta^{(1)},\theta^{(2)})=\mathcal{V}(s^{(t)};\theta^{(1)})+\mathcal{G}(s^{(t)},a^{(t)};\theta^{(2)}) \tag{33}$$

由此可以看出，给定 $\mathcal{Q}$ 值不可能唯一获取 $\mathcal{V}$ 和 $\mathcal{G}$，导致式性能的不确定性. 为了解决该问题并提高算法稳健性，将式(33)重写为式(34). 式(34)的优点在于并未改变深度 $\mathcal{Q}$ 学习算法框架，仅需获取优势函数的平均值，而无需找到状态 $s^{(t)}$ 下所有可能动作的最大值.

$$\mathcal{Q}\left(s^{(t)},a^{(t)};\theta^{(1)},\theta^{(2)}\right)=\mathcal{V}\left(s^{(t)};\theta^{(1)}\right)+\mathcal{G}\left(s^{(t)},a^{(t)};\theta^{(2)}\right)-\frac{1}{|\mathcal{A}|}\sum_{a^{(t+1)}\in\mathcal{A}}\mathcal{G}\left(s^{(t)},a^{(t+1)};\theta^{(2)}\right) \tag{34}$$

由于最初的深度强化学习框架是为视频游戏开发的[18]，因此这种架构包含了卷积层来处理输入层的图像. 本文旨在解决无蜂窝毫米波大规模MIMO系统中的资源优化问题，因此使用的神经网络结构更简单，仅包含两个全连接结构的隐藏层[13, 19]，以捕获系统当前工作的特定状态. 本文的值函数和优势函数均由有两个具有800个神经元的隐藏层的全连接网络构成，即两个隐藏层 H_1 和 H_2，一个值函数输出层Lv以及一个优势函数输出层La来分别估计值函数 $\mathcal{V}(s;\theta^{(1)})$ 和优势函数 $\mathcal{G}(s,a;\theta^{(2)})$，两者合并后得到深度决斗网络的输出 $\mathcal{Q}(s,a;\theta^{(1)},\theta^{(2)})$. 假设 $|\mathrm{H}_i|$ 表示网络层的神经元个数，深度决斗神经网络复杂度为 $\mathcal{O}\left(N_{\mathrm{Duel}}N_b\left(|H_1||H_2|+|H_2||\mathrm{Lv}|+|H_2||\mathrm{La}|\right)\right)$，其中 N_{Duel} 为训练迭代次数，N_b 为训练样本数.

无蜂窝毫米波大规模MIMO系统在不同时刻收集样本 $\mathcal{D}_t$，信道的实时变化使相同 (s,a) 在不同时刻获得的即时奖励也有所差异，因此学习过程中可能出现样本偏差，即相同的 $s=s^{(t)}=s^{(t')},\forall t\neq t'$，存在 $\varsigma^{(t)}\neq\varsigma^{(t')}$. 另外，代理和真实环境之间交互的计算复杂度高且特别耗时. 为了解决这些问题，本文提出将样本 $\mathcal{D}_t$ 存储到缓存空间，当出现样本偏差时，使用缓存空间中历史状态对应的样本代替偏差样本. 由于样本索引维度非常大，本文引入哈希函数检索方法来加快检索速度. 先对状态 $s^{(t)}$ 进行哈希编码 $\boldsymbol{h}_b(s^{(t)})$，然后根据哈希码索引存取信息 $\left(s^{(t)},\varsigma^{(t)}\right)$，哈希函数 $\boldsymbol{h}_b(s^{(t)})$ 的定义及检索方法参见附录C.

无蜂窝毫米波大规模MIMO系统不断与变化的外界环境进行交互，本文利用无蜂窝毫米波大规模MIMO系统知识和优化模型，采用基于干扰感知技术计算SINR，提出的决斗深度强化学习AP开关算法简称为Dueling-DQN-SINR算法，算法结构如图2所示. 算法结构主要由三个部分构成，分别为通信环境、通信模块以及决斗深度强化学习模块. CPU作为强化学习代理完成通信模块和深度强化学习模块的交互，同时，CPU在与通信环境交互时，在无蜂窝毫米波大规模MIMO系统的前传链路上利用现有的通信协议收集数据，并提供给通信模块. 通信模块基于干扰感知技术计算效用函数，引入基于哈希检索技术的缓存策略来避免样本偏差，最后通过对效用函数的分级处理为决斗深度强化学习模块提供分级状态输入. 决斗深度强化学习模块

提取 AP 开关信息后,将学习到的动作反馈给通信模型.

Dueling-DQN-SINR 算法的实现流程如算法 2 所示. 在决斗深度强化学习的数据收集阶段,基于投影基向量集获取哈希表. 在决斗深度强化学习的训练阶段,使用干扰感知技术、哈希检索法和状态空间分级算法获取经验信息$\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$,使其在深度强化学习框架下有效地学习. 为了避免长期使用历史信息而无法实时适应环境变化,代理以间隔 T 为周期清理缓存空间. 相对于其他深度强化学习算法,算法 2 具有以下优点:首先,干扰感知技术采用低维功率测量信息代替高维 CSI 信息进行信令交互,降低了系统复杂度;其次,分级算法使强化学习的状态空间大小从 2^M 降低到 P,单步动作转移策略降低了动作空间大小,从而提高了强化学习算法收敛速度;最后,缓存策略既能降低时变环境中的样本偏差,又能减少与复杂环境不必要的交互给前传链路带来的压力.

算法 2　基于 SINR 的决斗深度强化学习 AP 开关算法

输入: 初始化 $\mathcal{Q}$ 网络权重参数 $\theta^{(1)},\theta^{(2)}$ 和目标 $\mathcal{Q}$ 网络权重参数 $\bar{\theta}^{(1)},\bar{\theta}^{(2)}$

输出: $\mathcal{Q}$ 网络参数和 $\hat{\mathcal{Q}}$ 网络参数

while $t<|D|$
　以概率 ε 随机选择动作 $a^{(t)}$
　执行动作 $a^{(t)}$ 得到即时奖励 $r^{(t)}$ 和状态 $s^{(t+1)}$
　根据式计算 $\varsigma^{(t)}$, 然后得到 $\mathcal{D}_t$
　使用算法 1 得到信息 $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$
　将信息 $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$ 存放于内存池 D 中
end while
计算 $\boldsymbol{Z}$ 矩阵得到主成分向量对应的哈希表
repeat
　以概率 ε 随机选择动作 $a^{(t)}$
　根据式计算 $\varsigma^{(t)}$, 然后得到 $\mathcal{D}_t$
　使用算法 1 得到 $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$, 更新 $\tilde{\mathcal{D}}$
　将信息 $\left(s^{(t)},a^{(t)},r^{(t)},s^{(t+1)}\right)$ 存放于内存池 $\mathcal{D}$ 中
　从内存池 $\mathcal{D}$ 中随机提取少量训练样本 $\left(s^{(j)},a^{(j)},r^{(j)},s^{(j+1)}\right)$
　基于式, 在输出层输出 $\mathcal{Q}$ 值
　执行梯度下降法更新网络参数 $\theta^{(1)},\theta^{(2)}$
　间隔一段时间设置 $\hat{\mathcal{Q}}=\mathcal{Q}$,更新 $\bar{\theta}^{(1)}=\theta^{(1)},\bar{\theta}^{(2)}=\theta^{(2)}$
　if 深度神经网络收敛
　　break
end repeat

4　仿真与实现

为了研究无蜂窝毫米波大规模 MIMO 系统 AP 开关算法的性能,本文采用城市微小区(Urban Microcellular, UMi)开放广场场景大规模 MIMO 信道模型[16],AP 和用户之间的最大距离是 50 m. 热噪声功率谱密度为 −174 dBm/Hz,系统噪声系数 $F=6$ dB,系统中心频率 $f_0=73$ GHz,带宽 $B_0=200$ MHz,AP 数 $M=20$, 用户数 $K=6$, $N_{\mathrm{AP,max}}=10$. QoS 约束 $r_{\min}=1$ bit/s/Hz,最大功率约束 $P_{\max}=1$ dBW,每个 AP 采用用户平均功率分配策略,即 $p_{m,k}=P_{\max}/K,\ \forall k=1,2,\cdots,K$. $P_{c,m}$ 的计算过程描

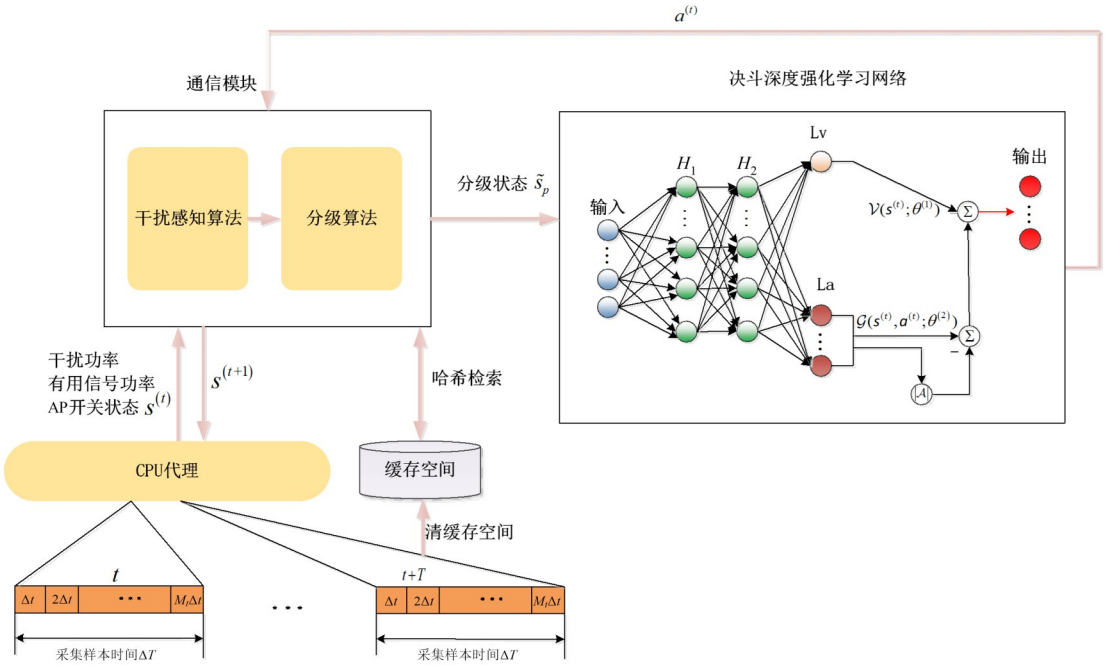


图 2　基于 SINR 的决斗深度强化学习 AP 开关算法结构

述参见文献[6,17],射频硬件电路功耗模型的具体参数参见表1,前传链路功耗参数参见文献[6].

表1 硬件模型参数

参数	含义	值
δ	功放效率	27%
P_{PS}	移相器功耗	21.6 mW
b_{DAC}	DAC分辨率	4 bit
P_{LO}	本地振荡器功耗	22.5 mW
P_{RF}	射频链功耗	31.6 mW
$P_{\mathrm{fh},0,m}$	前传链路固定功耗	5 W
$P_{\mathrm{fh},m,\mathrm{sleep}}$	前传链路睡眠固定功耗	0.5 W
$P_{\mathrm{fh},m}$	前传功率系数	0.25 W $\cdot$ Gbits$^{-1}\cdot$ s^{-1}

Dueling-DQN-SINR算法实现分别由两个平台完成:Python语言平台进行决斗深度强化学习算法仿真,Matlab平台进行通信模块和通信环境的仿真. 分级算法的离散度抽样间隔为$1/P$,$P=100$. 决斗深度神经网络参数为$|H_1|=800$,$|H_2|=800$,$|\mathrm{Lv}|=1$,$|\mathrm{La}|=M+1$,深度神经网络输入状态$s^{(t)}$的数据长度为M,内存池大小$|D|=2\,000$,训练样本数$N_b=64$,目标$\hat{\mathcal{Q}}$网络每隔200次训练更新一次. 学习率$v^{(t)}=0.001$,探索阶段随机动作概率ε从1逐步递减为0.1,长期奖励折扣因子$\gamma=0.9$.

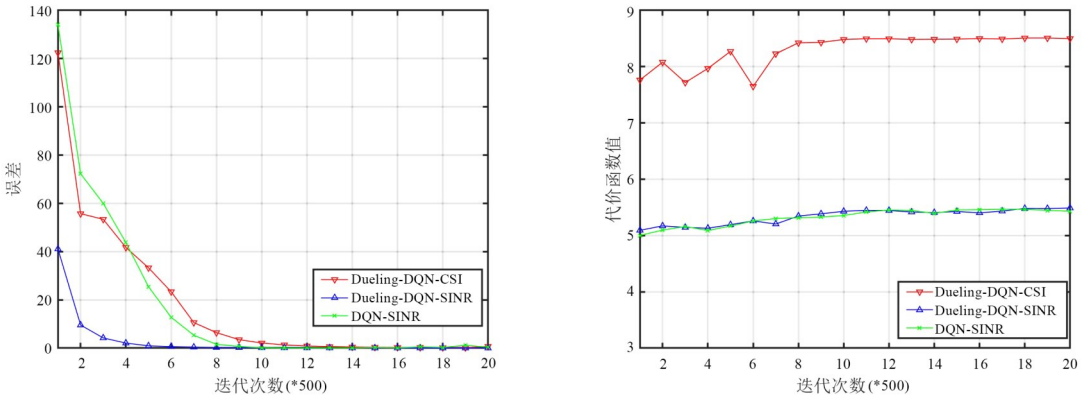
本文仿真比较了几种AP开关算法的性能:Dueling-DQN-SINR算法、基于SINR的DQN算法(DQN-SINR算法)、基于CSI的Dueling-DQN算法(Dueling-DQN-CSI算法)、文献[6]的能效贪婪算法和文献[3]的可达速率贪婪算法. 其中Dueling-DQN-SINR算法为推荐算法,其实现过程如算法2所示. DQN-SINR算法和Dueling-DQN-CSI算法均采用迷宫问题模型. 不同于Dueling-DQN-SINR算法中Dueling-DQN网络采用了两个隐藏层,DQN-SINR算法采用了一个隐藏层的DQN网络. Dueling-DQN-CSI算法采用了Dueling-DQN网络,但未采用分级操作,代理学习打开、关闭或者不变三种动作后,使用下一时刻CSI信息采用遍历搜索法选择总能效最大的AP. 能效贪婪算法考虑了QoS约束,采用Gauss-Seidel迭代算法来最大化总能效,其算法收敛性与初值的选取有很大关系,因此是一种次优算法. 可达速率贪婪算法采用贪婪策略最大化用户最小可达速率.

图3(a)给出了三种深度强化学习算法训练阶段的收敛性,图中横坐标中(*x)表示每点取x次迭代的平均值. 结果显示Dueling-DQN-CSI算法收敛速度最慢,这是由于其超大$\mathcal{Q}$表(大小为3×2^M)使学习过程难以收敛. Dueling-DQN-SINR算法因采用了分级操作,$\mathcal{Q}$表大小减小为$(M+1)\times P$,其收敛速度较快. Dueling-DQN-SINR算法采用了比DQN-SINR算法更复杂的网络结构,因而具有更快的收敛速度,DQN-SINR算法中DQN网络训练复杂度为$\mathcal{O}\left(N_{\mathrm{DQN}}N_b\left(|H_2||\mathrm{Lv}|+|H_2||\mathrm{La}|\right)\right)$. 仿真显示Dueling-DQN-SINR算法在6 000次迭代后逐渐收敛,其收敛速度比其他深度强化学习算法快一倍,即$N_{\mathrm{Duel}}=6\,000$,$N_{\mathrm{DQN}}=12\,000$. 图3(b)给出了深度强化学习算法在训练阶段的效用函数值,效用函数值随着迭代次数的增加而逐步增长. 基于干扰感知技术和CSI技术的深度强化学习算法的效用函数值有较大差异,后者通过准确估计$\mathcal{Q}$表能缩小总能效可行域范围. 经过一定时间的样本更新,根据统计信息获得Dueling-DQN-SINR算法的总能效可行域范围为[15,33],最大频谱效率为$r_{\max}=R_{\max}/B_0=2\ \mathrm{bit}\cdot\mathrm{s}^{-1}\cdot\mathrm{Hz}^{-1}$;Dueling-DQN-CSI算法的总能效可行域范围为[28,33],最大频谱效率为$r_{\max}=1.75\ \mathrm{bit}\cdot\mathrm{s}^{-1}\cdot\mathrm{Hz}^{-1}$.

图4仿真了基于干扰感知技术的深度强化学习算法在训练阶段中的平均性能. 在6 000次迭代之后,Dueling-DQN-SINR算法的平均总能效性能优于DQN-SINR算法,且用户平均最小频谱效率更接近QoS约束,这是由于前者考虑了值函数的处理,避免对不必要动作进行估计,因而能更有效地学习.

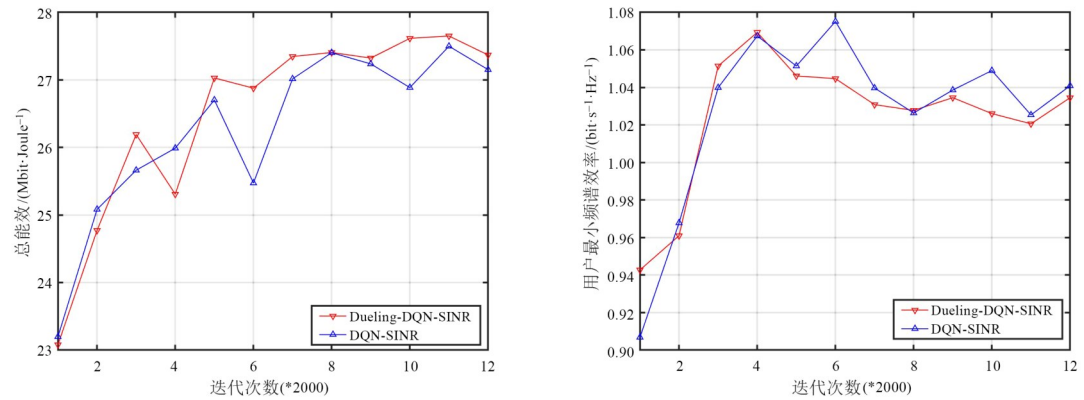
图5仿真了决斗深度强化学习算法和非强化学习算法在训练阶段的平均性能,采样样本大小$M_t=200$. Dueling-DQN-CSI算法的权衡性能表现最佳,虽然Dueling-DQN-SINR算法的平均总能效性能略低于Dueling-DQN-CSI算法,但是它不需要提前获取CSI信息,能通过学习选择AP,因而是一种更实用的方法. 能效贪婪算法由于没有强化学习的探索过程扰动,其性能表现相对平稳,但正是因为缺乏探索过程而无法进一步提升平均总能效性能. 频谱效率和总能效是互相矛盾的性能指标,本文旨在确保实现QoS约束下的总能效最大化,相对于能效贪婪算法,Dueling-DQN-SINR算法虽然牺牲了少量频谱效率性能,但是其平均总能效更高. 比如在6 000次迭代之后,Dueling-DQN-SINR算法的平均总能效提升了8%~21%,而用户平均最小可达速率仅降低了0~6%,且仍满足QoS约束,从而证明了本文推导的效用函数的有效性

图6仿真了深度强化学习算法在测试阶段的瞬时性能和平均性能,采样样本大小$M_t=10$. 图6(a)和(b)展示了深度强化学习算法前200次迭代的瞬时性能,图6(c)和(d)展示了前600次迭代的平均性能. 由于以可达速率最大化为优化目标,可达速率贪婪算法的用户平均最小可达速率性能最好,但是其总能效性能最低. 由图6(a)可知,由于Dueling-DQN-CSI算法的动作



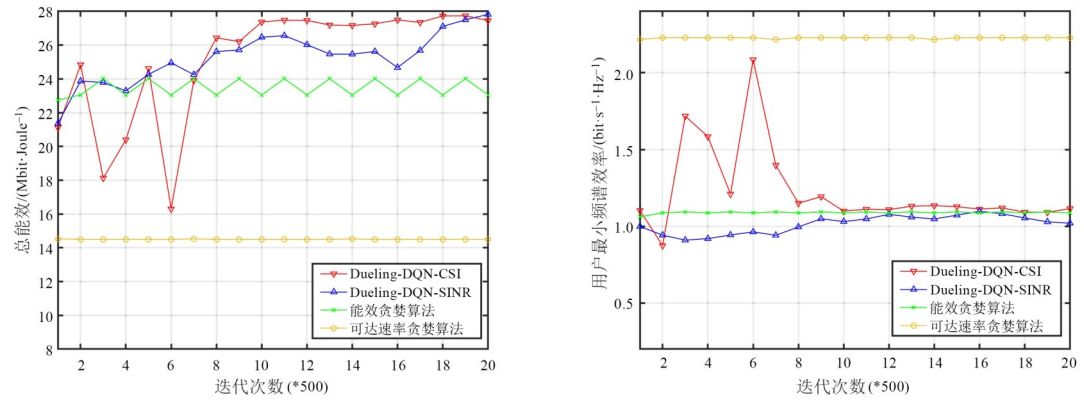
(*a*) 收敛性　　(*b*) 代价函数

图 3　深度强化学习算法训练阶段的收敛性和效用函数



(*a*) 平均总能效　　(*b*) 用户平均最小频谱效率

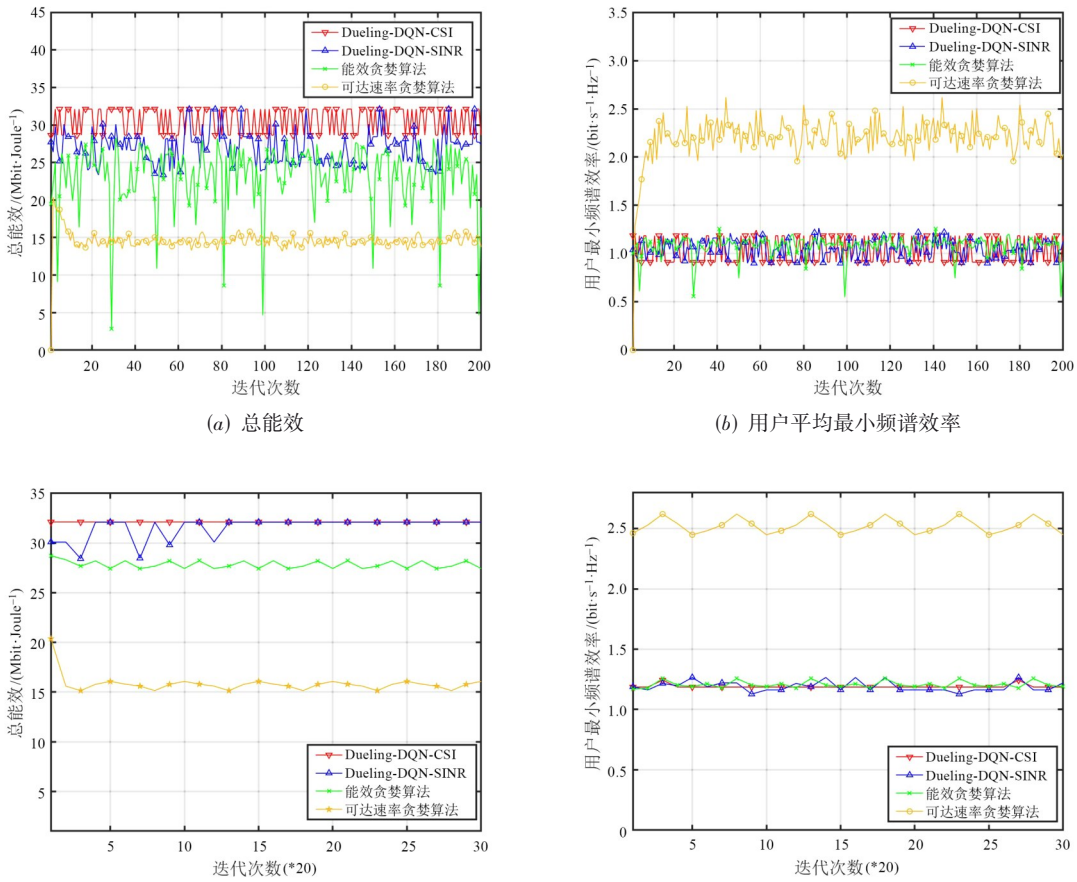
图 4　深度强化学习算法训练阶段的性能对比



(*a*) 平均总能效　　(*b*) 用户平均最小频谱效率

图 5　AP开关算法训练阶段的性能对比

空间小,其性能波动范围小;由于Dueling-DQN-SINR算法采用了分级处理,其瞬时总能效在高性能区间内波动. 本文在深度强化学习单次迭代中采用 $\tilde{r}_{\min}=r_{\min}-\Delta r$ 代替 $r_{\min}$ 来松弛QoS约束,确保用户在一段时间内而非在某一时刻满足QoS约束,其中 $\Delta r=0.1\ \text{bit}\cdot\text{s}^{-1}\cdot\text{Hz}^{-1}$. 图 6(*b*)给出单次迭代中用户最小频谱效率下限约为 $0.9\ \text{bit}\cdot\text{s}^{-1}\cdot\text{Hz}^{-1}$,图 6(*d*)给出用户平均最小频谱效率约为 $1.2\ \text{bit}\cdot\text{s}^{-1}\cdot\text{Hz}^{-1}$,满足QoS约束. 深度强化学习算



(*a*) 总能效 (*b*) 用户平均最小频谱效率

(*c*) 平均总能效 (*d*) 用户平均最小频谱效率

图 6 AP开关算法测试阶段的性能对比

法相对于能效贪婪算法取得了更高的平均总能效性能，比如在图6(*c*)中第15×20次迭代之后，深度强化学习算法的平均总能效提高了14%~16%. 图6仿真说明深度强化学习算法采用的单步长动作转移策略能使瞬时性能随着时变信道动态变化，经过一定步长动作转移后，两种深度强化学习算法的平均总能效均能达到较高稳定值，且满足QoS约束.

5 结论

本文研究了无蜂窝毫米波大规模MIMO系统基于睡眠机制的能效优化策略，主要通过关闭某些AP来降低硬件和前传链路的功率. 采用了决斗深度强化学习算法估计AP开关状态，以优化QoS约束下的总能效性能. 本文提出了一种低复杂度的效用函数构造方法，该方法基于干扰感知技术能在严格QoS约束下实现总能效和可达速率性能权衡. 提出缓存策略来减少样本误差和信令交互，并引入哈希检索法提高检索效率. 针对状态空间大导致的深度强化学习收敛慢问题，提出将效用函数进行分级处理以减小神经网络输入的状态空间大小. 仿真表明推荐的深度决斗学习算法相对于其它算法具有收敛快的特点，能在满足QoS约束下提高总能效性能.

附录A

性质1的证明考虑两种极限情况：

1. 无QoS约束的最大总能效定义为$\mathcal{K}_{\max}$，则效用函数ς_1为

$$\varsigma_1 = -\mu + (1-\mu)\psi(\mathcal{K}_{\max}) \tag{A1}$$

2. 最大可达速率$R_{\max}$对应的总能效定义为$\mathcal{K}_{\min}$，则效用函数ς_2为

$$\varsigma_2 = \mu + (1-\mu)\psi(\mathcal{K}_{\min}) \tag{A2}$$

在满足QoS约束时，最优总能效表示为$\mathcal{K}^*$，则对应的效用函数ς^*为

$$\varsigma^* = (1-\mu)\psi(\mathcal{K}^*) \tag{A3}$$

为了找到最接近$R_{\min}$时最大总能效，即最优μ能满

足式,则需满足$\varsigma^* > \varsigma_1$且$\varsigma^* > \varsigma_2$,即

$$\psi(\mathcal{K}_{\max}) - \psi(\mathcal{K}^*) < \frac{\mu}{1-\mu} < \psi(\mathcal{K}^*) - \psi(\mathcal{K}_{\min}) \quad \text{(A4)}$$

即需要满足

$$\frac{\psi(\mathcal{K}_{\max}) + \psi(\mathcal{K}_{\min})}{2} < \psi(\mathcal{K}^*) \quad \text{(A5)}$$

由于$\psi(\mathcal{K})$是凸函数,则式(A5)成立. 为了保证最优能效值附近的效用函数连续,则效用函数也需要满足$\psi(\mathcal{K}_{\max}) > \psi(\mathcal{K}^*)$,$\mu/(1-\mu)$取值为两个边界的中值,即

$$\frac{\mu}{1-\mu} = \frac{\psi(\mathcal{K}_{\max}) - \psi(\mathcal{K}_{\min})}{2} \quad \text{(A6)}$$

即μ满足定义.

附录B

基于马尔科夫过程[20, 21]的强化学习在代理和环境之间的交互可以表示为一个轨迹,即

$$\tau = s^{(1)}, a^{(1)}, r^{(1)}, s^{(2)}, a^{(2)}, r^{(2)}, \ldots, s^{(T)}, a^{(T)}, r^{(T)} \quad \text{(B1)}$$

强化学习代理根据马尔可夫随机策略$\pi: \mathcal{S} \to \mathcal{A}$来选择动作,$\psi(a|s)$为根据策略$\pi(s)$在状态$s$中选择动作$a$的概率,则轨迹$\tau$的概率用马尔科夫过程表示为

$$p(\tau) = p(s^{(1)}) \prod_{t=1}^{T} \psi(a^{(t)} | s^{(t)}) p(a^{(t+1)} | s^{(t+1)}) \quad \text{(B2)}$$

其中,p为分布概率. 每个代理基于给定策略π在状态$s^{(t)}$采取动作$a^{(t)}$后得到的即时奖励为

$$r^{(t)}(s^{(t)}, a^{(t)}) = r^{(t)}(s^{(t)}, \pi(s^{(t)})) \quad \text{(B3)}$$

定义$\mathcal{V}^{\pi}(s): \mathcal{S} \to \mathbb{R}$为值函数[20],表示在初始状态$s \in \mathcal{S}$采用策略$\pi$的期望值. 定义状态动作值函数为$\mathcal{Q}(s,a)$函数,它表示初始状态为$s$和动作$a$采用策略$\pi$的期望值.

$$\begin{aligned} \mathcal{V}^{\pi}(s) &= \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r^{(t)}(s^{(t)}, a^{(t)}) \middle| s_0 = s \right] \\ &= \mathrm{E}_{\pi}\left[r^{(t)}(s^{(t)}, a^{(t)}) + \gamma \mathcal{V}^{\pi}(s^{(t+1)}) \middle| s_0 = s \right] \end{aligned} \quad \text{(B4)}$$

$$\begin{aligned} \mathcal{Q}^{\pi}(s,a) &= \mathrm{E}_{\pi}\left[r^{(t)}(s^{(t)}, a^{(t)}) + \gamma \mathcal{V}^{\pi}(s^{(t+1)}) \right] \\ &= r^{(t)}(s^{(t)}, a^{(t)}) + \gamma \mathrm{E}_{\pi}\left[\mathcal{V}^{\pi}(s^{(t+1)}) \right] \end{aligned} \quad \text{(B5)}$$

其中,s_0表示初始状态,γ表示长期奖励的折扣因子. $R(\pi)$为给定策略π的平均奖励,$R(\pi)$不依赖于初始状态. 马尔科夫过程的目标是找到一个最优策略π^*,以使决策的未来奖励最大,即

$$\max_{\pi} R(\pi) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathrm{E}\left(\gamma^t r^{(t)}(s^{(t)}, \pi(s^{(t)})) \right) \quad \text{(B6)}$$

使用每个状态的最优动作获取最优状态动作值函数,即

$$\mathcal{V}^{\pi^*}(s) = \max_{a^{(t)} \in \mathcal{A}} \left\{ \mathrm{E}_{\pi}\left[r^{(t)}(s^{(t)}, a^{(t)}) + \gamma \mathcal{V}^{\pi}(s^{(t+1)}) \right] \right\} \quad \text{(B7)}$$

对于所有(s,a)对,最优$\mathcal{Q}$函数由下式取得:

$$\mathcal{Q}^{\pi^*}(s,a) = r^{(t)}(s^{(t)}, a^{(t)}) + \gamma \mathrm{E}_{\pi}\left[\mathcal{V}^{\pi}(s^{(t+1)}) \right] \quad \text{(B8)}$$

从式和式可知,最优$\mathcal{V}^{\pi^*}(s)$可表示为

$$\mathcal{V}^{\pi^*}(s) = \max_{a \in \mathcal{A}} \left\{ \mathcal{Q}^{\pi^*}(s,a) \right\} \quad \text{(B9)}$$

对于每个(s,a),最优$\mathcal{Q}$值可通过迭代方式实现[13],即

$$\begin{aligned} \mathcal{Q}^{(t+1)}(s^{(t)}, a^{(t)}) &= \mathcal{Q}^{(t)}(s^{(t)}, a^{(t)}) + v^{(t)}\left[r^{(t)}(s^{(t)}, a^{(t)}) \right] \\ &\quad + v^{(t)}\left[\gamma \max_{a^{(t+1)} \in \mathcal{A}} \mathcal{Q}^{(t)}(s^{(t+1)}, a^{(t+1)}) - \mathcal{Q}^{(t)}(s^{(t)}, a^{(t)}) \right] \end{aligned} \quad \text{(B10)}$$

式(B10)的原则是要找到当前估值$\mathcal{Q}^{(t)}(s^{(t)}, a^{(t)})$和目标$\mathcal{Q}$值$r^{(t)}(s^{(t)}, a^{(t)}) + \gamma \max_{a^{(t+1)} \in \mathcal{A}} \mathcal{Q}^{(t)}(s^{(t+1)}, a^{(t+1)})$之间的差值,通过不断更新式对应的$\mathcal{Q}$表,算法可以逐渐收敛到最优策略. 其中学习率$v^{(t)}$表示新的经验对当前估计的$\mathcal{Q}$值的影响,学习率可在训练过程中进行调整,为了确保$\mathcal{Q}$学习算法始终收敛于最优策略,学习率必须是非负的、确定性的.

附录C

假设样本集构成的数据集矩阵为$\boldsymbol{U} = [u_1, u_2, \cdots, u_N] \in \mathbb{C}^{M \times N}$,$u_n$是数据集中一个特定的$M$维样本向量,$\boldsymbol{U}$的协方差矩阵为$\boldsymbol{\Sigma} = (\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}})/N$. $\boldsymbol{V}$为协方差矩阵$\boldsymbol{\Sigma}$的特征矩阵,选择矩阵$\boldsymbol{V}$中最大$\boldsymbol{L}$个特征值对应的特征向量作为基向量集,则新数据集$\boldsymbol{Y}$在基向量集$\boldsymbol{Z}$上的投影为$\boldsymbol{Y} = \boldsymbol{Z}^{\mathrm{H}}\boldsymbol{U}$. 由于$\boldsymbol{Z}$包含数据分布信息,因此使用$L$个投影向量$\boldsymbol{Z} = [Z_1, Z_2, \cdots, Z_L]$的哈希表也是面向数据的. 这$L$个投影向量对应$L$个哈希函数,每个哈希函数$h_{Z_i,b}(u_n): \mathbb{R}^M \to \mathbb{R}$, $1 \leqslant i \leqslant L$表示一个$M$维向量$u_n$的映射. 它由变量$Z_i$和$b$进行索引,$b$是从$[0,w]$范围内均匀选择的实数. 对于样本$u_n$,投影向量$Z_i$和$b$对应的哈希函数为:

$$h_{Z_i,b}(u_n) = \left\lfloor \frac{Z_i^{\mathrm{H}} u_n + b}{w} \right\rfloor \quad (51)$$

多个哈希值$h_{Z_i,b}(u_n)$, $1 \leqslant i \leqslant L$组成哈希表$\boldsymbol{h}_b(u_n): \mathbb{R}^M \to \mathbb{R}^L$, $\boldsymbol{h}_b(u_n) = \left[h_{Z_1,b}(u_n), h_{Z_2,b}(u_n), \cdots, h_{Z_L,b}(u_n) \right]$. 由于$L < M$,则哈希函数方法能降低高维度数据集的检索结构,从而能减少样本u_n的检索时间.

参考文献

[1] KIM S, SHIM B. Energy-efficient millimeter-wave cell-free systems under limited feedback[J]. IEEE Transactions on Communications, 2021, 69(6): 4067-4082.

[2] VAN CHIEN T, BJöRNSON E, LARSSON E G. Joint power allocation and load balancing optimization for energy-efficient cell-free massive MIMO networks[J]. IEEE Transactions on Wireless Communications, 2020, 19(10): 6798-6812.

[3] FEMENIAS G, LASSOUED N, RIERA-PALOU F. Access point switch ON/OFF strategies for green cell-free massive MIMO networking[J]. IEEE Access, 2020, 8: 21788-21803.

[4] USAMA M, EROL-KANTARCI M. A survey on recent trends and open issues in energy efficiency of 5G[J]. Sensors, 2019, 19(14): 3126.

[5] ZHUANG B, GUO D, HONIG M L. Energy-efficient cell activation, user association, and spectrum allocation in heterogeneous networks[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(4): 823-831.

[6] GARCíA-MORALES J, FEMENIAS G, RIERA-PALOU F. Energy-efficient access-point sleep-mode techniques for cell-free mmWave massive MIMO networks with non-uniform spatial traffic density[J]. IEEE Access, 2020, 8: 137587-137605.

[7] HE H, JIN S, WEN C-K, et al. Model-driven deep learning for physical layer communications[J]. IEEE Wireless Communications, 2019, 26(5): 77-83.

[8] PHAM Q V, MIRJALILI S, KUMAR N, et al. Whale optimization algorithm with applications to resource allocation in wireless networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4285-4297.

[9] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[10] HASHMI U S, ZAIDI S A R, IMRAN A, et al. Enhancing downlink QoS and energy efficiency through a user-centric Stienen cell architecture for mmWave networks [J]. IEEE Transactions on Green Communications and Networking, 2020, 4(2): 387-403.

[11] SUN G L, ZHAN T, OWUSU B G, et al. Revised reinforcement learning based on anchor graph hashing for autonomous cell activation in cloud-RANs[J]. Future Generation Computer Systems, 2020, 104: 60-73.

[12] SUN G L, AYEPAH-MENSAH D, XU R, et al. Transfer learning for autonomous cell activation based on relational reinforcement learning with adaptive reward[J]. IEEE Systems Journal, 2021, 16(1): 1044-1055.

[13] VAN HUYNH N, HOANG D T, NGUYEN D N, et al. DeepFake: Deep dueling-based deception strategy to defeat reactive jammers[J]. IEEE Transactions on Wireless Communications, 2021, 20(10): 6898-6914.

[14] DATAR M, IMMORLICA N, INDYK P, et al. Locality-sensitive hashing scheme based on p-stable distributions [C]//Proceedings of the twentieth annual symposium on Computational geometry. New York: ACM, 2004: 253-262.

[15] CHAFIK S, YACOUBI M A EL, DAOUDI I, et al. Unsupervised deep neuron-per-neuron hashing[J]. Applied Intelligence, 2019, 49(6): 2218-2232.

[16] ALONZO M, BUZZI S, ZAPPONE A, et al. Energy-efficient power control in cell-free and user-centric massive MIMO at millimeter wave[J]. IEEE Transactions on Green Communications and Networking, 2019, 3(3): 651-663.

[17] RIBEIRO L N, SCHWARZ S, RUPP M, et al. Energy efficiency of mmWave massive MIMO precoding with low-resolution DACs[J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(2): 298-312.

[18] WANG Z, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning[C]// International Conference on Machine Learning. New York: ICML, 2016: 1995-2003.

[19] SARWAR S S, SRINIVASAN G, HAN B, et al. Energy efficient neural computing: A study of cross-layer approximations[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2018, 8(4): 796-809.

[20] ZHANG T, ZHU K, WANG J. Energy-efficient mode selection and resource allocation for D2D-enabled heterogeneous networks: A deep reinforcement learning approach [J]. IEEE Transactions on Wireless Communications, 2020, 20(2): 1175-1187.

[21] ZHANG J, HUANG Y, WANG J, et al. Intelligent interactive beam training for millimeter wave communications

[J]. IEEE Transactions on Wireless Communications, 2020, 20(3): 2034-2048.

作者简介

何　云　女,1979年生,湖北武汉人,现为重庆邮电大学博士生研究生. 主要研究方向为协作通信、大规模MIMO系统能效优化.
E-mail: heyun@cqupt.edu.cn

申　敏(通讯作者)　女,1963年生,贵州湄潭人. 现为重庆邮电大学教授、博士生导师. 主要研究方向为通信核心芯片、协议与系统应用技术.
E-mail: shenmin@cqupt.edu.cn

王　蕊　女,1988年生,云南玉溪人. 现为重庆邮电大学博士生研究生. 主要研究方向为Cell-Free大规模MIMO系统的资源分配、动态协作和预编码.
E-mail: d190101012@stu.cqupt.edu.cn

张　梦　女,1988年生,重庆人. 现为浙江大学博士后. 主要研究方向为物理层安全,物联网安全.
E-mail: zhangmengyang@zju.edu.cn