• 人工智能 •

基于多智体强化学习的接入网络切片动态切换

秦 爽，赵冠群，冯 钢*

(电子科技大学通信抗干扰技术国家级重点实验室 成都 611731)

【摘要】网络切片技术将广泛应用于以 5G 为代表的下一代移动通信网络中，为网络中多样化的业务提供按需的网络服务。在基于切片的移动通信网络中，用户往往需要根据不断变化的网络状态，进行接入切片的动态切换，以获得更好的网络传输和服务性能。考虑到存在多个用户的网络中，某一用户的接入选择将对接入切片的可用传输资源产生影响，从而影响其他用户的接入和切换决策。因此，该文将基于网络切片的移动通信网络中多用户的接入切换建模为一个多人随机博弈问题，采用多智体强化学习的方法对该问题进行求解，并设计了一种基于分布式多智体强化学习算法的多用户接入切片动态切换机制。在此基础上，通过仿真实验验证了该切换算法性能。

关 键 词 接入切换； 多智体强化学习； 多人随机博弈； 网络切片

中图分类号 TN929.5 **文献标志码** A doi:10.12178/1001-0548.2020049

Dynamical Accessing Handoff by Using Multi-Agent Reinforcement Learning in Slice Based Mobile Networks

QIN Shuang, ZHAO Guan-qun, and FENG Gang*

(National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China Chengdu 611731)

Abstract In future mobile networks, such as 5G networks, network slicing will be a promising technology to provide customizing services for different users with different transmission requirements. According to the dynamic network state in slice based mobile networks, users need to make accessing slice handoff periodically for improving the transmission performance. However, in a multi-user networks, the accessing choice of a user changes the amount of available transmission resources in the system, which impacts the accessing choices of other users. Thus, in this paper, we model the multi-user handoff problem in slice based mobile networks as a multi-agent random game. Then, we use multi-agent reinforcement learning (MARL) to solve this game, and propose a multi-user accessing handoff algorithm based on distributed MARL method. The numerical results validate the performance of our proposed multi-user accessing handoff algorithm in slice based mobile networks.

Key words accessing handoff; MARL; multi-agent random game; network slices

以 5G 为代表的未来移动通信系统，将广泛采用 SDN 和 NFV 技术，通过构建软件定义的移动通信网络，为用户提供灵活按需的网络传输服务[1-2]。在软件定义的移动通信网络架构下，将采用网络切片技术，来满足多样化业务的差异化服务需求，也越来越成为研究者的共识[2-4]。每个端到端网络切片服务于网络中某一类具有特定需求的业务，在逻辑功能层面对应相互独立的端到端虚拟网络，多个切片对应的不同逻辑虚拟网络，将通过映射部署到相同的物理网络之上。

当移动用户到达网络时，需要选择一个满足自身业务服务需求的切片接入网络。在实际的网络中，网络条件和用户业务需求动态变化，使得用户到不同接入站点的信道条件以及不同切片中的可用资源情况不断变化。因此，为了保证用户的接入和传输性能，需要根据用户的接入信道条件和可用资源情况，进行用户接入切片的动态切换。在传统的移动通信网络中，用户的接入切换只需要考虑从一个接入站点切换到另一个接入站点。而在基于切片的软件定义移动通信网络中，一个接入站点上往往

收稿日期：2020－01－20；修回日期：2020－02－15
基金项目：国家自然科学基金重点项目 (61631005)；广东省重点领域研发计划项目 (2018B010114001)
作者简介：秦爽 (1984－)，男，博士，副教授，主要从事无线及移动通信网络方面的研究.
通信作者：冯钢，E-mail：fenggang@uestc.edu.cn

部署了多个不同的网络切片，而同一切片可能覆盖多个不同的接入站点。由此，用户与接入站点二者之间的接入选择和切换问题，就变成了用户、切片和接入站点三者之间的优化匹配问题。

在移动通信网络中，用户的接入切换一直是研究热点[5-8]。但现有的研究主要关注传统移动通信网络中的用户切换问题，而对于如何在基于切片的软件定义移动通信网络中，进行用户接入切片的动态优化切换，保障用户业务的服务性能，还少有涉及。同时，在实际的通信系统中，切片的可用传输资源有限，接入同一切片的多个用户将竞争有限的传输资源。某一用户的接入选择，会改变接入切片中可用传输资源数量，进而对其他用户的接入和传输性能产生影响。因此，需要综合考虑网络中多个用户的接入决策之间的相互制约和影响关系，从提升多个用户整体传输性能的角度，设计多用户协同的接入切片动态切换机制。

本文重点关注了基于网络切片的软件定义移动网络中，移动用户接入切片的动态优化选择和切换问题。首先，考虑到多个用户共存的网络中，不同用户的接入选择将相互影响相互制约，结合移动通信应用场景下，动态的网络条件和业务需求对用户接入决策的影响，将网络中多个用户的接入切换建模为一个多人随机博弈问题。然后，通过多智体强化学习 (multi-agent reinforcement learning, MARL) 方法[9-11] 对该问题进行求解，并提出了一种基于分布式多智体强化学习[12] 的多用户接入切换算法。在此基础上，通过仿真实验，验证本文提出算法的性能。

1　系统模型

本文考虑的网络模型如图 1 所示，M 个基站组成的移动网络中部署了 N 个网络切片。多个切片部署在相同的物理网络之上，共享相同的物理传输资源，包括接入网的无线传输带宽和功率，以及核心网的传输带宽。一个切片可能覆盖多个基站，一个基站上也可能部署多个不同的切片，基站的无线传输资源将根据需求被分配给各个切片。在接入网，多个不同的基站之间可以通过 Xn 接口相互连接，各个基站通过 NG 接口连接到核心网中的 AMF(access and mobility management Function)。AMF 负责切片的部署和管理，一个 AMF 可以同时管理多个切片。AMF 通过与 SDN 控制器的信息交互，可以获得切片在核心网中可用的传输资源情况，并通过 NG 接口告知部署了该切片的各个基站。而一个用户可能处于多个接入站点的覆盖范围内，由此通过基站的广播信息，可以获得不同基站上可接入的切片状态信息，并从中选择合适的切片接入。为了便于分析，本文假设一个用户只产生一条业务流，用户和业务是一一对应关系。

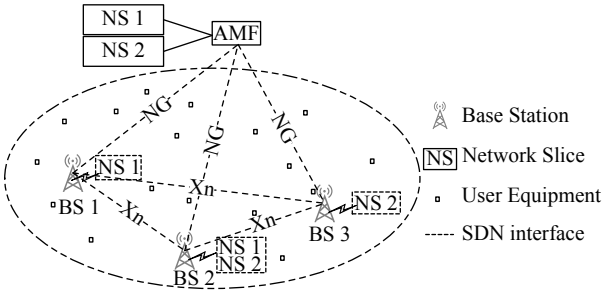

图 1　网络模型

当一个用户接入某一切片，其获得的服务速率，同时受切片在接入网和核心网中可用传输资源的限制。如果用户通过基站 m 上的切片 n 接入网络，则其在接入端获得的无线接入速率 $r_{m,n}$ 将由用户到基站 m 的信道条件，以及此时切片 n 在站点 m 上可用的无线传输带宽和传输功率共同决定。而用户在核心网能够获得传输速率 c_n，则由切片 n 在核心网部分的容量以及业务负载共同决定。因此，用户接入网络后可以获得的端到端服务速率 $b_{m,n} = \min(r_{m,n}, c_n)$。由于如何进行切片传输资源的优化配置并非本文的关注重点，为了便于分析，本文简单假设切片覆盖范围内的用户信道条件相近，且切片可用的接入网和核心网传输资源平均分配给接入切片的多个用户。因此，用 $b_{m,n}^{\max}$ 表示站点 m 上的切片 n 能够提供的总的最大传输速率，则当有 K 个用户同时通过站点 m 上的切片 n 接入网络时，某一用户 k 获得服务速率 $b_{m,n}^{k} = b_{m,n}^{\max}/K$。

由于基站 m 通过与 AMF 的交互，可以获得切片 n 核心网部分的容量和负载。结合切片 n 在基站 m 获得的无线传输资源数量，基站可以得出当前基站 m 上的切片 n 能够提供的最大传输速率 $b_{m,n}^{\max}$。同时，根据当前接入用户数量，基站就可以计算出当前有新用户 k 接入网络时能够提供的服务速率 $b_{m,n}^{k}$，并且可以通过基站广播，把这一信息提供给用户。

2　问题建模

2.1　多用户切换策略

网络用户的移动及业务需求的变化，使得各个

切片中服务的业务流不断到达和离开。由于每个切片在接入端和核心网可用的传输资源有限，切片负载的变化使得接入用户获得的服务速率动态变化。由此，考虑网络中的用户每隔一段时间，将根据当前网络状态的变化，判断是否需要进行接入切片的切换，以获得更高的服务速率。

用户在进行接入切换时，会带来相应的信令传输、处理时延等切换开销，频繁的切换可能导致过大的切换开销，从而降低用户体验和网络服务性能。因此，本文的设计目标是希望在通过用户接入切片的优化切换来提升用户服务速率的同时，尽量减少由此带来的切换开销。

2.2 多用户随机博弈模型

在实际网络中，往往多个用户同时处于多个切片的覆盖范围内，每个切片可用的传输资源有限，一个用户的接入选择，将对其他用户的可用传输资源和接入决策产生影响。因此，本文将动态网络条件下，多用户的周期性接入选择和切换决策过程建模为一个多用户随机博弈问题。

多用户随机博弈可以看作一个包含多个决策者的马尔科夫决策过程，并用元组$(S,A^1,A^2,\cdots,A^K,r^1,r^2,\cdots,r^K,p)$表示，其中 S 为系统状态空间，A^k为决策者 k 的动作空间，$r^k: S\times A^1\times\cdots A^K\to R$为回报函数，$p$ 为状态转移概率。本文考虑将每个用户看作一个决策者，在每个决策时刻，用户 k 执行动作 a^k，并且获得收益$r^k(s,a^1,a^2,\cdots,a^K)$。用户所处的状态 s 会根据转移概率跳转到下一状态 s’。

因此，在本文考虑的多用户切换对应的多人随机博弈问题中，每个用户周期性地根据当前网络状态，以最大化自身的累积收益为目标，进行接入切片的切换决策。多用户随机博弈问题中各个组成元素可以表示为：

1) 系统状态：用$s\in S$表示网络状态，其中 S 为所有状态的集合。设网络中存在 N 个切片，M 个基站，则某一时刻用户 k 所处的状态可表示为 $s^k=[I^k_{m,n},b^k_{1,1},\cdots,b^k_{m,n},\cdots,b^k_{M,N}]$。$I^k_{m,n}$表示决策者 k 当前通过基站 m 接入切片 n，反应了用户当前的连接状态。$b^k_{m,n}$的含义如第 1 节所述，表示当前时刻，用户 k 如果接入基站 m 上的切片 n 可以获得的服务速率。在实际网络中，往往很多基站上只部署了部分切片，因此如果基站 m 上没有部署切片 n 时，可以在状态向量中将对应的$b^k_{m,n}$去掉，从而降低状态空间的大小。同时，可以将传输速率表示为单位速率的倍数，通过将传输速率的离散化来进一步简化状态空间，则有$b^k_{m,n}\in B^k_{m,n}=\{1,2,\cdots,b^{\max}_{m,n}\}$，其中$b^{\max}_{m,n}$表示通过基站 m 上的切片 n 能提供的最大服务速率。

2) 动作：在本文所考虑的切换问题中，将用户的动作定义为用户对接入切片的选择。在每个决策周期，用户 k 采取动作$a^k=(m,n)$表示用户 k 选择通过基站 m 接入切片 n，其中$a^k\in A^k$，$A^k=\{(m,n)|1\leqslant m\leqslant N,1\leqslant n\leqslant M\}$为用户 k 的动作空间。

3) 状态转移概率：在多用户随机博弈过程中，每个决策时刻，网络中的 K 个用户同时执行所选取的动作，导致网络状态发生变化。因此，每个用户观察到的系统状态转移，将同时受其它用户采取的动作的影响，用户所处状态从 s 跳转到下一状态 s'的概率满足$\sum_{s'\in S}p(s'|s,a^1,a^2,\cdots,a^n)=1$。

4) 回报函数：在某一决策周期，用户 k 处于状态 s 采取动作a^k后，获得的立即回报由两部分决定：一是用户采取动作后获得的服务速率；二是用户接入切片发生切换所带来的切换开销。

值得注意的是，本文考虑切片将可用传输资源平均分配给接入的多个用户，所以用户 k 处于状态 s 采取动作a^k后获得服务速率与系统跳转后的状态相关，受系统中其他用户的动作影响。因此用$f^k(s,a^1,a^2,\cdots,a^K)$来表示在某一决策周期，处于状态 s 的用户 k 在所有用户采取联合动作$(a^1,a^2,\cdots,a^K)$后获得的服务速率。为了便于分析，假设用户获得单位服务速率的收益为 1。同时，当用户 k 采取动作a^k后，从状态$s=[I_{m,n},B_{1,1},\cdots,B_{m,n},\cdots,B_{M,N}]$跳转到状态$s'=[I_{m',n'},B'_{1,1},\cdots,B'_{m,n},\cdots,B'_{M,N}]$，用户的接入切片可能发生变化，从而带来切换开销$g^k(s,a^k)$。开销函数$g^k(s,a^k)$定义为：

$$g^k(s,a^k)=\begin{cases}K_{I_{m,n},I_{m',n'}} & I_{m,n}\neq I_{m',n'}\\ 0 & I_{m,n}=I_{m',n'}\end{cases}\tag{1}$$

式中，$K_{I_{m,n},I_{m',n'}}$表示用户接入状态从$I_{m,n}$切换到$I_{m',n'}$时带来的切换开销，其值往往由网络中执行接入切换带来的信令传输、处理时延等多项因素共同决定。为了便于分析，本文假设其是一个常数，即$K_{I_{m,n},I_{m',n'}}=K_c$。

在多用户随机博弈过程中，用户之间的决策会相互影响。因此考虑每个用户都是以最大化系统的累积收益为目标进行切换策略的优化决策，从而将用户的立即回报函数定义为 K 个用户的总收益，即：

$$r^k(s,a^1,a^2,\cdots,a^K)=\sum_{k=1}^{K}(f^k(s,a^1,a^2,\cdots,a^K)-g^k(s,a^k))\tag{2}$$

在多人随机博弈问题中，如果所有决策者都具有相同的回报函数，则称为团队博弈。已有研究证明，在团队博弈中，存在全局最优均衡点[14]。本文采用多智体强化学习 (MARL) 方法来求解上述多人随机博弈问题。

3　模型求解与算法设计

3.1　MARL 方法

多人随机博弈可以看作一个多智体强化学习问题。在包含 K 个智能体的 MARL 中，设智能体 k 的策略为π^k，则根据文献 [13]，其状态值函数可以表示为：

$$v^k(s,\pi^1,\pi^2,\cdots,\pi^K)=E\left[\sum_{t=0}^{\infty}\gamma\cdot r_t^k|\pi^1,\pi^2,\cdots,\pi^K,s_0=s\right] \tag{3}$$

式中，γ为折扣因子；r_t^k为用户 k 在决策时刻 t 获得的立即回报。

与传统强化学习相比，MARL 存在多个智能体，在求解对应的多用户随机博弈问题时，可以将传统的 Q-Learning 方法[15] 扩展到多智体系统。对于一个 K 个智能体构成的多智体系统，对应的 Q 函数可以表示为：

$$\begin{aligned}&Q^k(s,a^1,a^2,\cdots,a^{\mathrm{K}})=r^k(s,a^1,a^2,\cdots,a^K)+\\&\alpha\sum_{s'\in S}p(s'|s,a^1,a^2,\cdots,a^K)v^k(s',\pi^1,\pi^2,\cdots,\pi^K)\end{aligned} \tag{4}$$

式中，α为探索率，$(a^1,a^2,\cdots,a^K)$和$(\pi^1,\pi^2,\cdots,\pi^K)$分别为 K 个智能体的联合动作和联合策略；$r^k(s,a^1,a^2,\cdots,a^K)$为用户 k 的立即回报，可由式 (2) 得到。

本文考虑的多用户切换问题中，在每个决策时刻，一但用户的联合动作$(a^1,a^2,\cdots,a^K)$确定，则 K 个用户的连接状态就确定了，由此可以确定系统的跳转状态s'，并得到$p(s'|s,a^1,a^2,\cdots,a^K)$=1。则式 (4) 可以简化为：

$$\begin{aligned}&Q^k(s,a^1,a^2,\cdots,a^K)=r^k(s,a^1,a^2,\cdots,a^K)+\\&\alpha v^k(s',\pi^1,\pi^2,\cdots,\pi^K)\end{aligned} \tag{5}$$

由此，对应的多智体 Q-Learning 算法中，Q 函数的更新公式可表示为：

$$\begin{aligned}&Q^k(s,a^1,a^2,\cdots,a^K)\leftarrow(1-\alpha)Q^k(s,a^1,a^2,\cdots,a^K)+\\&\alpha[r^k(s,a^1,a^2,\cdots,a^K)+\gamma\max_{a_*^1,a_*^2,\cdots,a_*^K}Q^k(s',a_*^1,a_*^2,\cdots,a_*^K)]\end{aligned} \tag{6}$$

3.2　基于分布式多智体 Q-Learning 的切换算法

在基于网络切片的软件定义移动通信网络中，利用 SDN 控制器，可以方便地实现集中控制的多智体 Q-learning 算法。由式 (6) 可以看到，集中控制算法中，Q 值函数与所有用户的联合动作$(a^1,a^2,\cdots,a^K)$相对应，这使得算法的状态空间和动作空间都较大，导致很高的算法复杂度。因此，本文考虑采用一种分布式的在线多智体 Q-Learning 算法[12]，每个智能体只维护与自身动作相对应的 Q 值函数，降低了算法的复杂度，同时算法运行过程中用户之间只需进行少量的信息交互。

本文设计的分布式 Q-Learning 算法如下。在该算法中，每次迭代计算，智能体 k 根据当前的网络 s，独立地采取$\varepsilon-$greedy策略选择自己的动作a^k。由此，可以得到网络中的联合动作$(a^1,a^2,\cdots,a^K)$。执行动作后，智能体通过观察网络转移到的新状态 s’计算得到的立即回报$r^k(s,a^1,a^2,\cdots,a^k)$，并更新对应的动作值函数$Q^k(s,a^k)$。

算法 1　基于分布式 Q-learning 的动态切换决策算法

输入：S; A; r; α; γ；

输出：策略向量$(\pi_*^1,\pi_*^2,\cdots,\pi_*^K)$

1) 初始化$Q^k(s,a^k)=0$，$\forall a^k\in A_k,k=1,2,\cdots,K$

2) Repeat

3) 获取当前 s

4)　if exploration then

5)　随机选择$a^k\in A_k$，$k=1,2,\cdots,K$

6)　if exploitation then

7)　$a^k=\arg\max_a Q^k(s,a)$，$k=1,2,\cdots,K$

8)　for k=1,2$\cdots$,K

9)　观察下一状态 s'，agent k 获得的回报 $r^k(s,a^1,a^2,\cdots,a^K)$

10)　$Q^k(s,a^k)\leftarrow(1-\alpha)Q^k(s,a^k)+\alpha[r^k(s,a^1,a^2,\cdots,a^K)+\gamma\max_{a_*^k\in A}Q^k(s',a_*^k)]$

11)　$s\leftarrow s'$

12)　end for

13) until(完成特定步数或所有$Q^k(s,a^k)$都收敛)

值得注意的是，分布式算法中，每个智能体只需要维护与自身动作相对应的动作值函数$Q^k(s,a^k)$，而不需要维护联合动作值函数$Q(s,a^1,a^2,\cdots,a^K)$。但这并不表示在算法中，每个智能体完全独立地进行学习。由系统状态的定义$s^k=[I_{m,n}^k,b_{1,1}^k,\cdots,b_{m,n}^k,\cdots b_{M,N}^k]$可知，当智能体要判断当前所处状态时，需要获得网络切片当前可以提供的服务速率。这除了取决于智能体自身的接入选择决策外，也将受其他智能体接入选择策略的影响。此外，从算法第 10) 行可以

看到，Q 函数的更新需要获得联合动作$(a^1,a^2,\cdots,a^K)$下的立即回报$r^k(s,a^1,a^2,\cdots,a^K)$。因此，为了计算立即回报，在此多智体系统中，智能体之间需要通过基站进行必要的信息交互。

3.3 算法复杂度

在本文的多用户随机博弈问题中，系统状态空间的大小为$|S|$，每个用户的动作空间大小为$|A|$，设用户的数量为 K。则可以得到，在对应的分布式 Q-Learning 算法执行过程中，系统中所有智能体需要维护的 Q 值表中状态-动作对的总数为$K\cdot|S|\cdot|A|$。因此，在算法运行过程中，存储所有 Q 值表所需要的存储空间复杂度和算法每次迭代运算的计算复杂度都是$K\cdot|S|\cdot|A|$。

与本文中采用的分布式 Q-learning 算法相比，传统的多智体算法中，动作值函数由所有智能体的联合动作决定，表示为$Q(s,a^1,a^2\ldots,a^K)$，则每个智能体对应 Q 值表中的状态-动作对的个数就变为了$|S|\cdot|A|^K$。因此，系统中所有智能体需要维护的 Q 值表中状态-动作对的总数就是$K\cdot|S|\cdot|A|^K$。因此，传统的多智体 Q-learning 算法运行过程中，空间复杂度和每次迭代的计算复杂度为$K\cdot|S|\cdot|A|^K$。相比于传统的多智体 Q-learning 算法，算法 1 采用的分布式多智体 Q-learning 算法在计算复杂度和空间复杂度上都有明显的提升。

4 数值结果分析

在仿真实验中，考虑将设计的 MARL 算法与多种传统算法性能进行对比，对比算法包括：

1) Fixed 算法：用户在到达网络后，固定选择一个切片接入，不进行切换；

2) RSS-based 算法：在每个决策时间点，用户总是选择 RSS 最大的基站上的切片接入；

3) BW-based 算法：在每个决策时间点，用户总是选择能够提供最大服务速率的切片接入；

4) SAW (simple additive weighting method) 算法：用户仅考虑自己采取的动作带来的收益，不考虑用户之间的相互影响。在每个决策时间点，选择收益最大的切片接入。

在图 1 网络场景下进行仿真实验，仿真参数如表 1 所示。假设每个基站覆盖范围内有业务不断动态到达或离开，业务的到达和离开服从泊松分布，对应的联合到达速率可以表示为$\lambda=(\lambda_1,\lambda_2,\cdots,\lambda_M)$，其中$\lambda_m$为基站 m 上的用户到达速率，同理有$\mu=(\mu_1,\mu_2,\cdots,\mu_M)$。本文中的数值结果为 100 次随机仿真结果的平均值。

表 1 仿真实验参数

仿真参数	参数定义	取值
M	基站数量	3
N	切片数量	2
UEs	用户数量	<8
$b_{m,n}^{\max}$	切片最大服务速率	U[0,10]
K_c	切换开销	1
α	学习率	0.2
ε	探索参数	0.1
γ	折扣因子	0.9
λ	到达速率	（2,1,1）
μ	离开速率	（1,2,1）

图 2 和图 3 分别给出了系统中的累积回报和吞吐量随决策步数的变化关系。如图所示，在不同的算法下，系统累积回报和吞吐量的值都随着决策步数的增加而递增，其中本文提出的 MARL 算法的性能总是优于其他算法。BW-based 算法和 RSS 算法分别根据服务速率最大和 RSS 最大进行切换决策，没有考虑切换开销带来的影响，可能导致较多的切换和较大的切换开销。Fixed 算法在用户接入网络后不进行切换，当网络条件发生变化时无法切换到性能更好的切片。而 SAW 算法在进行切换决策时，不考虑其他用户决策的影响，可能导致多个用户选择相同切片接入，从而竞争有限的切片资源。而 MARL 算法一方面综合考虑了用户服务速率和切换开销的之间相互影响和约束关系，另一方面也考虑了系统中多个用户间的相互竞争关系，因此能取得比其他几种算法更好的性能。

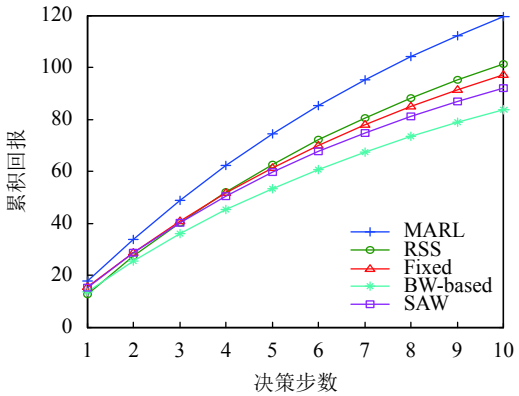

图 2 累积回报

图 4 对比了不同算法下系统中的累积切换次数。从图中可以看出，除了 Fixed 算法不会进行切

换，MARL 算法的切换次数少于其他几种算法。其中，RSS 算法的切换次数相对较少，这是因为仿真中没有考虑用户移动，用户到接入站点的信道条件相对固定，当用户找到信道强度较好的基站就基本不再切换。而 SAW 算法由于没有考虑其他用户接入选择的影响，容易造成多个用户竞争同一切片资源，导致切换次数较高。BW-based 算法只考虑了当前切片可以提供的服务速率，而没有考虑切换带来的开销，也会导致切换次数较高。

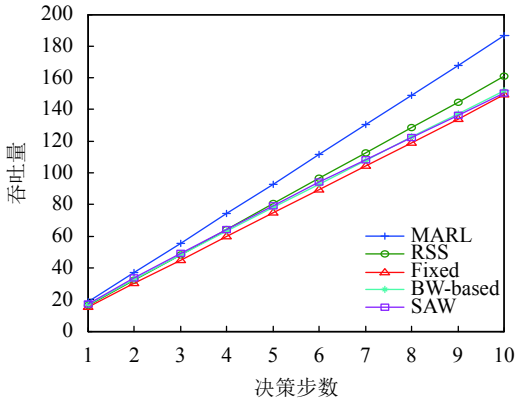

图 3　系统吞吐量

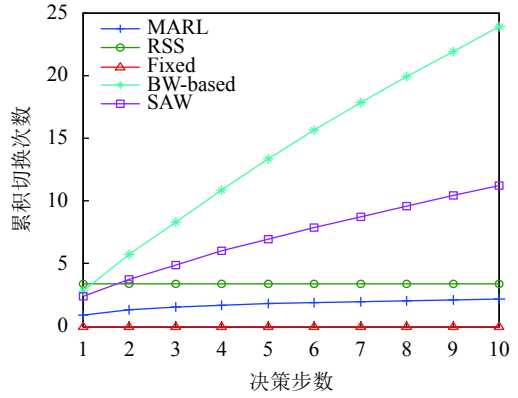

图 4　累积切换次数

图 5 给出了切换开销K_c与系统中累积回报的关系。随着K_c增大，各算法的累积回报逐渐减小。当K_c值较大时，如图$K_c>5$时，切换开销很大，用户通过切换获得的服务速率增益小于切换带来的开销，因此 MARL 算法下，用户基本不进行切换，算法曲线与 Fixed 算法重合。同理，SAW 算法在切换开销较大时也很少切换，使得累积回报基本保持不变。而 BW-based 算法在进行切换决策时并没有考虑切换开销的影响，因此不会因为K_c的增大额而调整自己的切换策略，使得其累积回报受K_c影响较大，随着K_c的增大而持续下降。

图 6 给出了随着切换开销的增大，不同切换算法下，系统吞吐量的变化情况。由于K_c较大时，为了避免较多的切换开销，MARL 算法倾向于较少的切换，使得很多用户不会切换到当前能够提供最大服务速率的切片，造成系统吞吐量下降。同理，SAW 算法吞吐量的变化规律与 MARL 算法类似，同样随着K_c的增大而降低。而其他 3 种算法在进行切换决策时，没有考虑切换开销的影响，随着K_c的增大，系统吞吐量基本不受影响。综合图 2～图 6 可以看到，与其他算法相比，本文提出的 MARL 算法能获得较好的网络传输和服务性能。

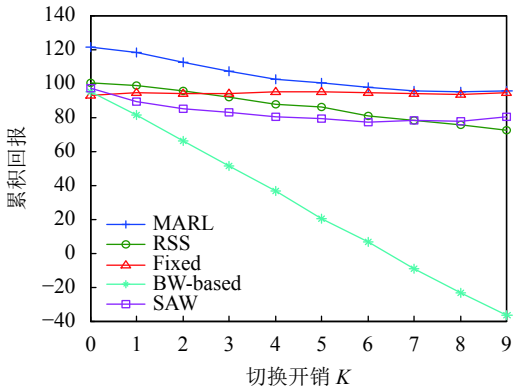

图 5　累积回报 vs.切换开销

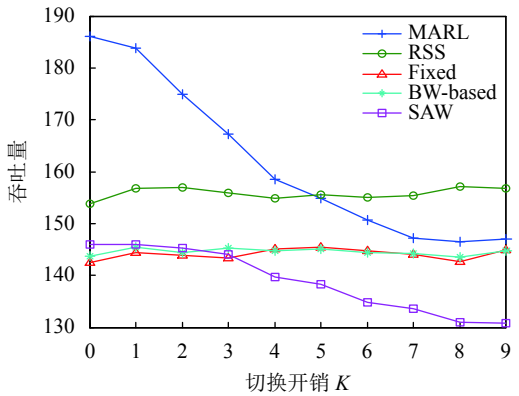

图 6　系统吞吐量 vs.切换开销

最后，图 7 和图 8 给出了本文提出的 MARL 算法的收敛性能。如图 7 所示，当α=0.2时，算法在进行约 20 000 次迭代训练后逐渐收敛。图中结果显示，α的取值越大，算法的收敛速度越快，但相应的数值结果波动越大，反之亦然。在此基础上，图 8 给出了所采用的分布式 Q-Learning 算法中，对应 Q 值函数的收敛情况。$Q(s,a_i)$ 为系统处于状态 s 时，采取动作 a_i 得到的动作值函数。从图中可以看到，与图 7 相似，在经过约 20 000 次迭代训练后，Q 函数的取值趋于稳定。

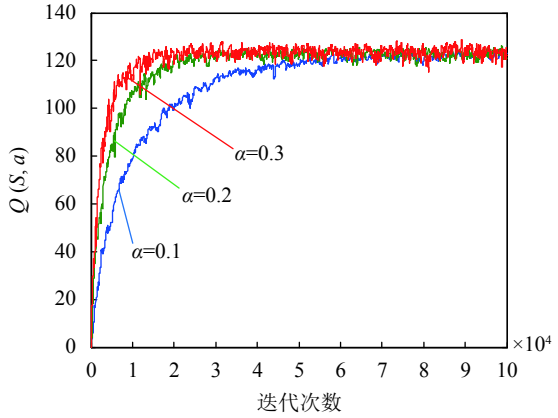

图 7 算法收敛速度

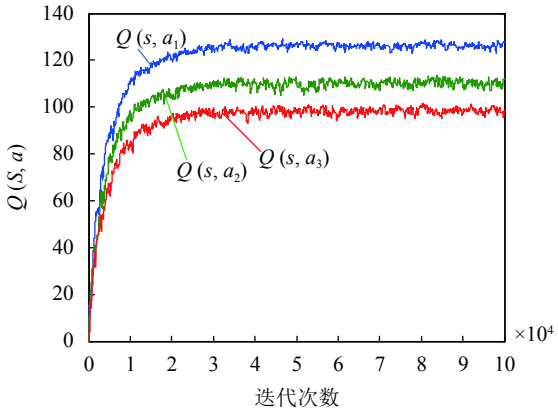

图 8 Q 值函数收敛情况

5 结 束 语

本文研究了面向网络切片的移动网络中，移动用户接入切片的动态优化切换问题。考虑到网络中，多个用户之间的相互影响和制约关系，将多用户协同的接入切片切换过程建模为一个多人随机博弈问题。在此基础上，设计了基于多智体强化学习的多用户接入选择和切换算法。仿真实验的结果证明，本文提出的算法能够在提升网络服务性能的同时降低网络中的切换开销。

参 考 文 献

[1] MARSCH P, SILVA I D, BULAKCI O, et al. 5G radio access network architecture: Design guidelines and key considerations[J]. IEEE Communications Magazine, 2016, 54(11): 24-32.

[2] ORDONEZ-LUCENA J, AMEIGEIRAS P, LOPEZ D, et al. Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges[J]. IEEE Communications Magazine, 2017, 55(5): 80-87.

[3] A N, X, ZHOU C, TRIVISONNO R, et al. On end to end network slicing for 5G communication systems[J]. Transactions on Emerging Telecommunications Technologies, 2017, 28(4): 1-11.

[4] WANG G, FENG G, QUEK T Q, et al. Reconfiguration in network slicing optimizing the profit and performance[J]. IEEE Trans on Network and Service Management, 2019, 16(2): 591-605.

[5] ARANI A H, OMIDI M J, MEHBODNIYA A, et al. A handoff algorithm based on estimated load for dense green 5G networks[C]//GLOBECOM'15: Proceeding of the 2015 IEEE Global Communications. San Diego: IEEE Press, 2015: 1-7.

[6] SUN Y, FENG G, QIN S, et al. The SMART handoff policy for millimeter wave heterogeneous cellular networks[J]. IEEE Trans on Mobile Computing, 2018, 17(6): 1456-1468.

[7] ZGOU G, PETER L, GAO H. A network controlled handover mechanism and its optimization in LTE heterogeneous networks[C]//The 2013 IEEE Wireless Communications and Networking Conference. Shanghai: IEEE Press, 2013: 1915-1919.

[8] LEEM H, KIM J, SUNG D K, et al. A novel handover scheme to support small-cell users in a HetNet environment[C]//The 2015 IEEE Wireless Communications and Networking Conference. New Orleans: IEEE Press, 2015: 1978-1983.

[9] JIANG W, FENG G, QIN S, et al. Multi-agent reinforcement learning for efficient content caching in mobile D2D networks[J]. IEEE Trans on Wireless Communications, 2019, 18(3): 1610-1622.

[10] NOWE A, VRANCX P, HAUWERE Y M. Game theory and multi-agent reinforcement learning[M]. Reinforcement Learning. Berlin, Heidelberg: Springer, 2012.

[11] YAN M, FENG G, ZHOU J. Smart multi-RAT access based on multi-agent reinforcement learning[J]. IEEE Trans on Vehicular Technology, 2018, 67(5): 4539-4551.

[12] SAAD H, MOHAMED A, ELBATT T. Distributed cooperative Q-learning for power allocation in cognitive femtocell networks[C]//The 2012 IEEE Vehicular Technology Conference (VTC Fall). Quebec City: IEEE, 2012: 1-5.

[13] SHOHAM Y, POWERS R, and GRENAGER T. Multi-agent reinforcement learning: A critical survey[R]. Stanford: Stanford University, 2003.

[14] HU J and WELLMAN M P. Nash Q-learning for general-sum stochastic games[J]. Journal of Machine Learning Research, 2003, 4(6): 1039-1069.

[15] SZEPESVARI C. Algorithms for reinforcement learning[J]. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2010, 4(1): 1-103.

编 辑 蒋 晓