Research Article

A Distributed E-Cross Learning Algorithm for Intelligent Multiple Network Slice Selection

Guomin Wu,^{1,2} Guoping Tan¹ ,¹ and Defu Jiang¹

¹College of Computer and Information, Hohai University, 210098, China

²College of Information Engineering, Yancheng Institute of Technology, 224051, China

Correspondence should be addressed to Guoping Tan; gptan@hhu.edu.cn and Defu Jiang; surfer_jiangdf0801@163.com

Received 15 July 2020; Revised 20 March 2021; Accepted 29 April 2021; Published 17 May 2021

Academic Editor: Telmo Reis Cunha

Copyright © 2021 Guomin Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, some technological issues in network slicing have been explored. However, most works focus on the physical resource management in this research field and less on slice selection. Different from the existing studies, we explore the problem of intelligent multiple slice selection, which makes some effort to dynamically obtain better user experience in a changeable state. Herein, we consider two factors about user experience: its throughput and energy consumption. Accordingly, a distributed E-cross learning algorithm is developed in the multiagent system where each terminal is regarded as an agent in the distributed network. Furthermore, its convergence is theoretically proven for the dynamic game model. In addition, the complexity of the proposed algorithm is discussed. A mass of simulation results are presented for the convergence and effectiveness of the proposed distributed learning algorithm. Compared with greedy algorithm, the proposed intelligent algorithm has a faster convergence speed. Besides, better user experience is attained effectively with multiple slice access.

1. Introduction

Nowadays, different user demands ought to be met in the various scenarios, such as high system capacity, high reliability, low latency, and high resource efficiency. With the diversity of these requirements, the existing network cannot provide various end-to-end services with a physical network [1–3]. To achieve the goal, some iron tower companies have been established all over the world, such as China, England, and America. In this architecture, several operators can operate different systems in the same physical network. As a key enabling technology in the emerging network, network slicing can render customized service for terminals in a flexible way [4]. This technology takes advantage of software defined network (SDN) and network function virtualization (NFV) to build several logical networks [5]. According to different demands, the whole physical network can serve different vertical sectors with various logical instances. Subsequently, each virtual instance can provide the service for users in the different scenarios, smart city, autonomous driving, virtual reality, etc., whereas some challenges appear in this technol-

ogy field, autonomous and dynamic network configuration, slice security, radio access network slicing, user equipment (UE) slicing, and others. Herein, one of the challenges is that mobile terminals are required to intelligently access to multiple logic slices on demand instantaneously in some cases [6]. Thus, it is meaningful to explore the problem about smart multiple slice selection.

The architecture of network slicing is made up of three following layers: physical resource layer, function layer, and service layer. At present, most of related studies focused on resource management in the physical layer. To efficiently manage physical resource, Reference [7] designed a distributed iterative algorithm to improve spectrum efficiency. In [8], an iterative algorithm was proposed to jointly optimize physical power allocation with fine granularity. Both the aforementioned research works were based on the iterative method to improve network performance [7, 8]. A corresponding scheme with virtual resource management was proposed to meet user requirements in [9]. To cope with resource division between slices, an approach was proposed for power allocation and user association. In this paper, the

research problem was modelled as convex optimization problem. Afterwards, an intelligent corresponding algorithm was designed with Lyapunov principle [10]. In [11], a scheme was presented for non-NOMA slice resource allocation in the three types of classic applications: enhanced mobile broadband (eMBB), massive machine type communications (mMTC), and ultrareliable low latency communications (URLLC). These aforementioned References [10, 11] proposed corresponding schemes to manage non-NOMA resource in the network slicing architecture. To improve network efficiency, Reference [12] proposed a priority-based resource scheduling algorithm. In [13], a slicing framework was revealed to improve network performance. Reference [14] studied the scheme with traffic load and wireless resource constraint. In [15], resource slicing in the wireless access network was regarded as a challenge. Afterward, four related schemes were proposed for slice management to maximize their resource efficiency. Reference [16] revealed a resource scheduling scheme to predict future network state, which was based on deep learning and reinforcement learning algorithm. In [17], a heuristic algorithm was developed for spectrum allocation, access control, and space multiplexing. Reference [18] proposed a resource slicing scheme in the wireless access network. As to resource slicing, dynamic programming was applied to model slice isolation, which could provide better network performance [19]. Reference [20] proposed a novel scheme to dynamically allocate network slice. These abovementioned papers [19, 20] studied resource allocation schemes in the virtual access network. In [21–28], various game models were used to efficiently manage wireless resource. Reference [21] made use of coalition game model to represent spectrum cooperation between operators. In [22], a distributed algorithm was proposed to allocated resource and service selection, which was modelled with match game. A network framework was designed for resource allocation, access control, and packet loss among the different slices [23]. Reference [24] modelled resource management as congesting game. Correspondingly, two distributed algorithms were revealed to allocate physical resource. In [25], Nash equilibrium was obtained to present resource allocation algorithm. Broken game model was used for spectrum management in the cloud access network [26]. Reference [27] modelled distributed resource allocation as game model. In [28], auction game was used to optimize network performance. Herein, all the abovementioned algorithms [7–28] were proposed for optimal physical resource management, which aimed to serve terminal or operator better.

As to slice selection, related work appeared to be few in the past [29–34]. In [29], slice selection was regarded as a vital part in the network slicing architecture. A genetic algorithm was proposed to find the optimal solution. Reference [30] firstly presented related principle about network slicing. Afterwards, a slice selection algorithm was given to improve network performance. An event-triggered slice selection mechanism was proposed in the end-to-end slice network [31]. In [32], the related slicing principle and work flow were shown in the paper. Reference [33] presented a joint optimization algorithm about virtual base station access and slice selection. Reference [34] presented some approaches about

slice selection. The aforementioned references covered some ranges of slice selection. However, these solutions are not applicable for our research problem. It is the reason that these models are not capable of formalizing the complicated matching relationship for terminal and slice: each terminal tries to obtain higher user experience, which considers its throughput and energy consumption. In addition, virtual network slice instances ought to provide better service for terminal user. Moreover, the network continuously changes their state from time to time. In other words, the network state is dynamically changeable in the whole process. Motivated by the aforementioned research gap, this paper mainly addresses the issue of intelligent multiple slice selection between terminals and multiple slices.

Herein, the following contributions are made in this paper.

- (1) In the whole network, each user ought to maximize its experience all the time. The Markov game is used to formulate dynamical selection among different slices. Herein, the user experience is presented as network utility, and the instantaneous user experience is regarded as a reward
- (2) Afterwards, a distributed E-cross learning algorithm is proposed to solve aforementioned multiple slice selection problem (SS_ECL), which considers two factors: its throughput and energy consumption. In this algorithm, its probability of slice has been dynamically changed over time
- (3) Finally, numerical results validate that the network can converge to be stable in the SS_ECL algorithm. In addition, user can obtain better experience, compared with greedy algorithm

The rest of this paper is organized as follows. The second section describes the scenario and presents the corresponding problem formulation. Afterwards, the third section shows the SS_ECL algorithm. The fourth section presents numerical results and corresponding analysis. Finally, the fifth section concludes the paper.

2. System Model and Problem Formulation

2.1. System Model. In Figure 1, various logic slices are mapped to several physical base stations. Afterwards, each virtual slice provides particular service for corresponding terminals. Base stations act as the bridge between terminals and slices. In other words, base stations are access points, meanwhile slice is regarded as the server for corresponding terminals. In addition, various slices have their own distinguished service priorities. According to these priorities, these terminals select their required slices. Three areas exist in this figure, which are called areas 1, 2, and 3, respectively. For the reason that terminals are mobile in the area, its received quality of service is changeable all the time. Afterwards, terminal dynamically accesses to other slices for better experience on demand.

FIGURE 1: System model.

Each terminal chooses its own slice according to its own experience. If the terminal obtains better quality of service, corresponding user will improve the probability of this access with the same strategy later. Otherwise, the user will reduce its probability. Therefore, the access process is applicable to reinforcement learning. Furthermore, the whole network consists of several users which change their access strategy independently. In detail, each user acts as an agent in this multiagent network. Every user cannot take some strategy for other agent, which means that it cannot control the slice access with other agent. In other words, each user agent only raises its own demand for serving slices. As a whole, all the agents provided various services with various slice instances in the same physical network. Thus, the whole network can be regarded as a distributed learning system.

2.2. Problem Formulation. As to the abovementioned multiple slice selection, the corresponding Markov model is presented as follows, which is formulated with the tuple $\langle S, A, P, R \rangle$ [35, 36].

- (1) S : network state set in the different time slots, which is donated by $S = \{s_1, s_2, \dots, s_T\}$. Herein, T is the index for maximum time slot
- (2) A : terminals choose their slices with the particular access strategies. In accordance with user experience, the access action set k is obtained as follows {slice 1, slice 2, slice 3, slice 1-2, slice 1-3, slice 2-3, slice 1-2-3}
- (3) P : in the particular network state, terminal dynamically changes its probability of slice selection when its user experience is variable. Herein, the higher probability prefers to slice selection with better user experience
- (4) R : terminal can obtain the instantaneous user experience after its slice selection

Besides, the corresponding utility function is derived as follows.

According to Shannon channel capacity formula, its data rate for the user is defined as follows:

$$TP_i = B_i \log \left(1 + \frac{Pt_x * g}{N} \right). \quad (1)$$

In Equation (1), TP_i is denoted as the channel capacity from a virtual slice i , B_i is the allocated spectrum band from the serving base station. In addition, N is denoted as the white Gaussian noise. Pt_x is denoted as the transmission power from its serving base station, and g is the channel gain between terminal and serving base stations.

Besides, the received power for terminal PC_i is expressed as follows:

$$PC_i = g * d^{-\lambda} * Pt_x. \quad (2)$$

Herein, d is denoted as the distance between them. λ is the parameter of path loss between them.

When the terminal attains some services from different slices, its throughput and power consumption is obtained as follows:

$$TP_{\text{sum}} = \sum_{i=1}^n TP_i, \quad (3)$$

$$PC_{\text{sum}} = \sum_{i=1}^n PC_i. \quad (4)$$

In Equation (4), n is the total number of various slices which serve as a specific terminal. In this scenario, the user experience is made up of two abovementioned parts. Among them, data rate is regarded as the instantaneous network gain and power consumption is referred to as the network cost. Therefore, the user experience utility [37] is defined as follows:

$$U = f_1(TP_{\text{sum}}) - f_2(PC_{\text{sum}}). \quad (5)$$

Herein, $f_1(TP_{\text{sum}})$ is the linear function of TP_{sum} . In the similar way, $f_2(PC_{\text{sum}})$ represents the linear function of PC_{sum} .

3. The Proposed SS_ECL Algorithm

Cross learning algorithm is a special case of standard reinforcement learning [38], which shows the random process of slice selection in the uncertain network environment. In the different time slots, each terminal can obtain the particular required service based on corresponding probability distribution. Moreover, its probability sector for different slice access is presented as follows:

$$p(n) = (p_1(n), \dots, p_N(n)). \quad (6)$$

As to each terminal, it cannot know about other strategy and reward. Therefore, it is a kind of game with incomplete

```

1: Initialization: user position, maximum iteration, learning speed
2: For x=1:maximum iteration
3:   For n=1:Num_user
4:     Initialize: action(p)
5:     According to network utility, reward(p) is obtained
6:   End For
7:   For n=1 Num_user
8:     For a=1: Num_slice
9:       If(a=action(p))
10:        Crosslearning = e*reward*(1-p(n,a))
11:       Else
12:        Crosslearning = -e*reward *p(n,a)
13:       P(n,a)=p(n,a)+Crosslearning
14:       Normalize: p(n,a)
15:     End For
16:   End For
17: End For

```

ALGORITHM 1: A distributed E-cross learning algorithm for intelligent multiple network slice selection (SS_ECL).

TABLE 1: Network parameter list.

Network parameter	Value
User number	500
Utilization ratio	[0.85, 0.7, 0.6]
Transmission power sector (dBm)	[20, 27, 30]
Allocated bandwidth (MHz)	10
Distance range (m)	rand [0, 200]

FIGURE 2: Convergent analysis for user ratio in the single slice access.

information. In each stage, every agent terminal updates its probability as follows:

$$p_i(n+1) = U_i + (1 - U_i)p_i(n) = p_i(n) + U_i(1 - p_i(n)), \quad (7)$$

$$p_j(n+1) = (1 - U_j)p_j(n) = p_j(n) - U_j p_j(n). \quad (8)$$

FIGURE 3: System throughput comparison in the single slice access.

In Equation (7), the process is shown for the update of user experience, which the probability with strategy i is increasing. In other words, the terminal can obtain better user experience with a positive reward. On the opposite, Equation (8) shows that the probability with the strategy j is decreasing, which means that the agent reduces the probability of this slice selection. It is the reason that the related negative reward has been presented for the agent.

Furthermore, the parameter e is developed to adjust its learning speed and improve corresponding network performance, which is described as follows:

$$p_i(n+1) = p_i(n) + eU_i(1 - p_i(n)), \quad (9)$$

$$p_j(n+1) = p_j(n) + eU_j(1 - p_j(n)). \quad (10)$$

FIGURE 4: Energy consumption comparison in the single slice access.

FIGURE 5: User experience comparison in the single slice access.

When the parameter e is large, the agent changes its probability of slice access fast. As a result, network convergence appears quickly. On the contrary, the whole network converges slowly when the parameter e is small. In addition, its error shows to be more when e is large, because cross learning implies discrete decision process for the terminal. With the terminal changing faster, the error of value easier tends to be more as a result.

Although the abovementioned equations have presented a solution for intelligent multiple network slice selection problem, it is still required to design an algorithm to present the executing entity for these equations. Thus, we propose a

distributed E-cross learning algorithm for intelligent multiple network slice selection in Algorithm 1.

Its convergence analysis for the SS_ECL algorithm is as follows. The user ratio for slices 1, 2, 3 is η_1, η_2, η_3 , respectively. In addition, these ratios ought to meet the following constraints $\eta_1 + \eta_2 + \eta_3 = 1$, $\eta_1, \eta_2, \eta_3 \in (0, 1)$. According to different service priorities, the logic relation for various slices is shown as follows:

$$U_1 > U_2 > U_3. \quad (11)$$

FIGURE 6: Learning speed effect on user ratio in the single slice access.

Considering Equation (9) and Equation (10), we obtain the following:

$$p_1(n+1) = p_1(n) + eU_1(1 - p_1(n)), \quad (12)$$

$$p_2(n+1) = p_2(n) - eU_2p_2(n), \quad (13)$$

$$p_3(n+1) = p_3(n) - eU_3p_3(n). \quad (14)$$

It is easy to know that the probability distribution for slice 1 exists in $(0, p_1)$. Similarly, the probability distribution for slice 2 exists in $(p_1, p_1 + p_2)$, and the probability distribution for slice 3 exists in $(p_1 + p_2, p_1 + p_2 + p_3)$. Besides, the whole distribution section is $(0, 1)$.

After several iterations, corresponding distributions are modified as follows:

$$(p_1, p_1 + p_2) = \phi, \quad (15)$$

$$(p_1 + p_2, p_1 + p_2 + p_3) = \phi, \quad (16)$$

$$(0, p_1) = (0, 1). \quad (17)$$

The corresponding analysis is as follows. In Equation (12), the probability is cumulative when the number of iterations is increased. Thus, corresponding function for slice 1 is monotonously increasing. However, the probability for slices 2 and 3 is reduced, respectively, in the similar way. Both corresponding functions show to be monotonously decreasing.

Furthermore, the user ratio for slice 1 is monotonously increasing. In addition, both of user ratios for slices 2 and 3 are monotonously decreasing.

FIGURE 7: Learning speed effect on system throughput in the single slice access.

Considering $\eta_1 + \eta_2 + \eta_3 = 1$, we obtain the following:

$$\lim_{n \rightarrow \infty} \eta_1 = 1, \quad (18)$$

$$\lim_{n \rightarrow \infty} \eta_2 = 0, \quad (19)$$

$$\lim_{n \rightarrow \infty} \eta_3 = 0. \quad (20)$$

Finally, most of the users tend to access the slice 1, which provides higher quality of service. Besides, few users choose other slices. Thereafter, the network converges to a stable state.

Besides, the complexity of the SS_ECL algorithm is discussed. As to the outer loop, K operations are necessary for Equation (2), and K is denoted as maximum iterations. For the inner loop, L operations are required for Equation (7), for the reason that the maximum number of terminals is L . Afterwards, M operation is necessary in Equation (8); in that, the total number of accessible slices reaches M . Hereafter, these corresponding parts are used to update the probability of slice selection, which operations can be negligible. In all, the total complexity of the SS_ECL algorithm is $O(K * L * M)$.

4. Numerical Results

To illustrate the convergence and effectiveness of the SS_ECL algorithm, greedy algorithm is selected as the baseline [39]. Considering different scenario requirements, terminal can choose a single slice or different slices instantaneously. In this simulation, base stations and users appear in the random way. Herein, Table 1 shows some related network parameters.

4.1. Network Performance Comparison in the Single Slice Access. Figure 2 presents the convergent diagram of single slice access for the SS_ECL and greedy algorithm. The learning speed is 0.0001, and the initial user ratio is 1/3 for each slice selection, respectively. As shown in Figure 2, both SS_ECL and greedy algorithm can converge to be stable step by step. The user ratio for slice 1 can tend to 1 with some continuous increment. Meanwhile, these user ratios for slice 2 and 3 decrease step by step and converge to be zero finally. It is the reason that both of aforementioned algorithms prefer to the particular slice with better user experience. With the iterations increasing, more and more users would like to select slice 1 in this scenario. In addition, there are some fluctuations in Figure 2. The corresponding analysis is as follows. The distance between terminal and serving base station is random. Furthermore, corresponding user experience is affected by this random distance. Thus, some performance fluctuation exists in this scenario.

In addition, convergent speed with greedy algorithm is faster than SS_ECL in the first 20 iterations. Later, SS_ECL overtakes the greedy algorithm in the convergent speed. The related analysis is shown as follows. Terminals with greedy algorithm prefer to select the slice with the maximum utility. However, SS_ECL can continuously update the probability of slice selection and obtain a better and better reward with the iterations increasing. Therefore, the convergent speed with SS_ECL algorithm is faster than the greedy algorithm later.

Figure 3 presents the throughput comparison with the greedy algorithm. It is obvious that the throughput is fairly proportional with the number of users for two algorithms. Some corresponding analysis is presented as follows. With the number of users increasing, the links between terminal and base station are incremental too. In addition, random learning process exists for these two algorithms. Each user selects its required slice probably. Thus, some performance fluctuation appears with it.

FIGURE 8: Learning speed effect on user energy consumption in the single slice access.

Figure 4 presents the energy consumption comparison in the single slice access. It is easy to find that system energy consumption is increased with the number of terminals. Nevertheless, two curves fluctuate a lot as a whole. It is due to the randomness of user location, which leads to random distance between serving base station and terminal. Moreover, the random path loss results in random power consumption. Therefore, some fluctuation exists in Figure 4.

Figure 5 presents the user experience comparison in the single slice access. As to the SS_ECL algorithm, the user experience for slice 1 tends to 0.37 after 50 iterations. In the meantime, user experience for other slice approaches to 0 in the first 50 iterations. For the greedy algorithm, the convergence of user experience for slice 1 reaches 0.37 after 200 iterations. The user experience for slice 2 or 3 converges to 0 step by step. The related analysis is presented as follows. The user experience consists of two parts: its throughput and energy consumption. In this scenario, most of users would like to choose the specific slice with better user experience. On the contrary, fewer users prefer to slice 2 or 3. With the iterations increasing, the user ratio approaches to 0. As a result, their corresponding user experience is reduced step by step.

In Figure 5, it is obvious that the convergent speed with the SS_ECL is faster than the greedy algorithm, because the SS_ECL algorithm can intelligently update its probability periodically. Better and better rewards are obtained for the users with the iterative number increasing. Consequently, the learning process speeds up convergent process in the SS_ECL algorithm. Thus, its convergent speed is faster than the greedy algorithm.

4.2. Learning Speed Effect in the Single Slice Access. Figure 6 presents the learning speed effect on the user ratio in the single slice access. When learning speed is 0.0001, corresponding user ratio of slice 1 approaches to 0.9. In the meantime, the user ratio for other slice tends to 0, respectively. In Figure 6, the user ratio shows to be convergent after the first 25 iterations. For the convergent difference, some reasons are shown as follows. When the learning speed is faster, the agent

FIGURE 9: Learning speed effect on user experience in the single slice access.

FIGURE 10: User ratio comparison in the multiple slice access.

user updates its probability of slice selection faster. In addition, each agent probably chooses the slice with higher network utility. Correspondingly, its convergent speed is faster finally.

Figure 7 shows the learning speed effects on the system throughput in the single slice access. It is obvious that the system throughput is roughly same, whether the learning speed is faster or not. In addition, some slight fluctuation exists in Figure 7, due to random distance between serving base station and user terminal. Besides, the SS_ECL is a learning algorithm with random Markov decision process, which leads to some uncertain probability of slice selection.

Figure 8 gives the learning speed effect on the energy consumption with more users. As we know, energy consumption

FIGURE 11: System throughput comparison in the multiple slice access.

exists in the each virtual link between serving base station and user. Due to the random distance between them, the energy consumption appears to be random. In addition, its deviation shows to be larger with more users, which results in larger curve fluctuation. Besides, its learning speed only affects the convergent speed, which is nothing with energy consumption.

Figure 9 presents the learning speed effect on the user experience in the single slice access. As shown in this figure, the user experience for slice 1 converges to 0.38 after 75 iterations when learning speed is 0.003. However, the convergent process begins after 210 iterations. In addition, it is easy to find that the faster learning speed results in faster

FIGURE 12: Energy consumption comparison in the multiple slice access.

FIGURE 13: User experience comparison in the multiple slice access.

convergence. Thus, the updating probability of slice selection is increased with learning speed. The SS_ECL can speed up with faster convergent speed as a result.

4.3. Network Performance Comparison in the Multiple Slice Access. Figure 10 presents the user ratio comparison in the multiple slice access. When the users in the service area have some requirements with high traffic, multiple slice access is necessary in this case. Herein, the learning speed is 0.0001. As shown in Figure 10, it presents the user ratio comparison with two algorithms in the multiple slice access. In other words, the terminal can access multiple slices instantaneously. In this figure, the user ratio for slices 1, 2, and 3 approaches to 1 at last. However, the user ratio for other slice access is reduced to 0 slowly.

In addition, the user ratio shows to be convergent after 70 iterations in the SS_ECL algorithm. Meanwhile, 200 itera-

tions are required to obtain the convergence in the greedy algorithm. Therefore, its convergence of the SS_ECL algorithm is faster than the greedy algorithm. The related analysis is presented as follows. The user prefers to the slice profile with higher utility to improve its quality of service in two algorithms. However, the system in the SS_ECL algorithm can intelligently modify its probability of slice selection in accordance with changeable network state. Thus, its network performance optimization is speeded up. Compared with the SS_ECL algorithm, its corresponding convergence appears to be gradual in the greedy algorithm. It is the reason that the greedy algorithm always tries to obtain the slice profile with the maximum utility. Furthermore, it is a lack of dynamical learning process in the network optimization.

Figure 11 presents the system throughput comparison with different users in two algorithms. It is obvious that system throughput is increased with more users in the two

algorithms. It is the reason that a slice shows a virtual link in which the corresponding data rate exists. With the iterations increasing, more users tend to make the particular choice with more slices.

Figure 12 presents the energy consumption comparison with various users. Firstly, it is easy to obtain that energy consumption shows to be greater when more iterations appear. Due to the larger number of users, more energy consumption is required. In particular, more data links exist in the whole network, which results in more energy consumption. Besides, the energy consumption shows to be random because the distance between user and its serving base station is random too. Therefore, the corresponding fluctuation appears in the network energy consumption.

In Figure 13, the initial user experience with two kinds of algorithms is 0.05 in all cases, respectively. With the iterative numbers increasing, the user experience for slice 1, 2, and 3 tends to 0.48 in the SS_ECL algorithm. Corresponding utilities converge to 0 with other choices of slice selection in the SS_ECL algorithm. The related analysis is presented as follows. Each user has the preference of choice with the higher user experience. Consequently, more and more users select the access profile with slices 1, 2, and 3 instantaneously, which can provide higher utility for them. On the other hand, the user experience with other slice access shows to be gradually reduced to 0.

As shown in Figure 13, it is obvious that only 35 iterations are required for convergence in the SS_ECL algorithm. In the meantime, the user convergence slowly appears after 210 iterations in the greedy algorithm. Its related explanation is presented as follows. With further iteration, the user continuously changes its strategy for slice access. A better and better reward is obtained for the users in the SS_ECL algorithm. However, the greedy algorithm always keeps the same strategy. Therefore, the speed of network optimization in the SS_ECL algorithm is faster than the greedy algorithm.

4.4. The Learning Speed Effect in the Multiple Slice Access.

Figure 14 presents the learning speed effect in the multiple slice access. The initial user ratio value is 0.15 when the learning speed is 0.001 or 0.002, respectively. As shown in Figure 14, the user ratio converges to 0.68 after 75 iterations when the learning speed is 0.001. However, only 40 iterations are necessary to obtain the convergence when the learning speed is 0.002. As to the gap of two convergent speeds, some related analysis is shown as follows. With the iterative numbers increasing, more increment of user ratio for slices 1, 2, and 3 is obtained when the learning speed is 0.002. As known, the probability is updated with a better reward, which increases in proportion with user experience.

Figure 15 presents the system throughput comparison in case of different learning speeds. With the number of users increasing, more links between base station and user appear in the service area. Consequently, system throughput is becoming greater and greater. It is the reason that the probability update of slice selection is in the learning process. Due to random user location, the performance fluctuation appears in Figure 15. With more users appear in the serving

FIGURE 14: Learning speed effect on user ratio effect in the multiple slice access.

FIGURE 15: Learning speed effect on system throughput in the multiple slice access.

network, the whole system throughput is increasing step by step.

Figure 16 presents the energy consumption in cases of different learning speeds. As a whole, the energy consumption is increased with more users. Its related analysis is shown as follows. More data links between serving base stations and users easily consume more energy in the service area. As to curve fluctuation in Figure 16, the similar analysis has been presented in Figure 12. Due to space limitation, the related part does not need to be analysed in detail.

Figure 17 shows the learning speed effect on the user experience in the multiple slice access. It is easy to find that both of user experiences are increased with more users when the learning speed is 0.001 and 0.002, respectively. Besides, only 25 iterations are necessary for performance convergence. However, it takes 70 iterations for the whole network

FIGURE 16: Learning speed effect on energy consumption in the multiple slice access.

FIGURE 17: Learning speed effect on user experience in the multiple slice access.

FIGURE 18: User ratio comparison in two cases.

state to be stable. Thus, the convergent speed shows to be faster when the learning speed is 0.002. Corresponding analysis is presented as follows. With the learning speed faster, the probability updating for slice access is faster.

4.5. Network Performance Comparison in Two Cases. Figure 18 shows the user ratio comparison in two cases, which include the single and multiple slice access. In the single slice access, most users access to network with slice 1 after 300 iterations. However, only 60 iterations are required for the network convergence in the multiple slice access case. Related analysis is presented as follows.

In the SS_ECL algorithm, its reward is directly proportional with network utility. In other words, the particular strategy with higher user ratio has more effect on its reward. In the single slice access, most of users prefer to slice 1 as an access point. It is the reason that slice 1 provides greater utility than other slice. In the multiple slice access, the selection

for slices 1, 2, and 3 is the preference for most users in the service area, because the slice profile can provide the best service quality for users. As shown in Figure 18, the solution with multiple slice access can provide more data links for the users than the case of single slice access instantaneously. Thus, the faster convergence can be found than the single slice access in Figure 18, which indicates that network performance shows to be better in the multiple slice access.

Figure 19 shows the system throughput comparison in two cases. With the number of users increasing, both of system throughputs are incremental. In addition, the system throughput with multiple accesses appears to be greater. Its related analysis is presented as follows. As known, a virtual link exists between a user and a slice. With the number of slices increasing, the whole throughput is increasing correspondingly.

Figure 20 presents the energy consumption comparison in two cases. With the increasing number of users, the system

FIGURE 19: System throughput comparison in two cases.

FIGURE 20: Energy consumption comparison in two cases.

FIGURE 21: User experience comparison in two cases.

energy consumption is incremental in two cases too. It is the reason that more slice accesses require more data links between user and serving base station, which tends to consume more energy in the network. In the similar way, the energy consumption with multiple slices shows to be more than that of single slice access. As to its curve fluctuation, it is on account of random path loss between serving base station and terminal.

Figure 21 shows the user experience comparison in two cases. It is obvious that 250 iterations are necessary for network convergence in the single slice access. However, it takes 75 iterations for the whole network to be convergent. In addition, the user experience with the multiple slice access shows to be better than that of the single slice access, because the former choice provides more data links for served user than the later at the same time. Thereafter, better user experience is obtained from multiple slice access finally.

5. Conclusions

In this paper, we have studied intelligent multiple slice selection to dynamically provide better service for various end-users instantaneously. First of all, the Markov game model is presented for the multiple network slice selection. Herein, user experience acts as the network payoff, and instantaneous user experience is regarded a reward correspondingly. Afterwards, an E-cross learning algorithm is proposed to solve the challenge, which considers two following factors: energy consumption and user throughput. The aforementioned probability is interactively updated to obtain better user experience. Simulation results demonstrate that the whole network can converge to be stable in the proposed SS_ECL algorithm and effectively attains better user experience. Nevertheless, the scenario with continuous state change shows to be more complicated and corresponding decision process shows to be uncertain, which is deferred as our future work.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61701168, 61832005, 61571303, and 61971179) and the Fundamental Research Funds for the Central Universities (No. 2014B33114). We specially thank Ziming Sheng, Shengjie Sun, Decun Wang, and Fengfan Jiang for their work in the simulation and others.

References

- [1] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, 2018.
- [2] 5G Americas, *5G Americas White Paper, Network Slicing for 5G and Beyond*, 2016.
- [3] J. Ordóñez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.
- [4] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.
- [5] China Unicom, *China Unicom White Paper, Network Slicing for 5G*, 2018.
- [6] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: a survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys & Tuts*, vol. 20, no. 3, pp. 2429–2453, 2018.
- [7] L. Tang, Y. Zhang, and R. Liang, "Virtual resource allocation algorithm for network utility maximization based on network slicing," *Journal of Electronics and Information Technology*, vol. 39, no. 8, pp. 1812–1818, 2017.
- [8] S. C. Lin, "End-to-end network slicing for 5G& B wireless software-defined systems," in *IEEE Global Communications Conference*, pp. 1–7, Abu Dhabi, United Arab Emirates, 2018.
- [9] N. Nikaein, C. Chang, and T. Spyropoulos, "Radio access network resource slicing for flexible service execution," in *IEEE Conference on Computer Communications Workshops*, pp. 668–673, Honolulu, HI, USA, 2018.
- [10] L. Tang, R. Ma, H. Yang, and S. Fu, "Joint user association and power allocation algorithm for network slicing based on NOMA," *Journal of Electronics and Information Technology*, vol. 41, no. 9, pp. 2039–2046, 2019.
- [11] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: a communication-theoretic view," *IEEE Access*, vol. 6, no. 9, pp. 55765–55779, 2018.
- [12] Q. Wang, J. Fu, J. Wu, B. Moran, and M. Zukerman, "Energy-efficient priority-based scheduling for wireless network slicing," in *IEEE Global Communications Conference*, pp. 1–6, Abu Dhabi, United Arab Emirates, 2018.
- [13] T. Guo and A. Suárez, "Enabling 5G RAN slicing with EDF slice scheduling," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2865–2877, 2019.
- [14] A. A. Gebremariam, M. Chowdhury, M. Usman, A. Goldsmith, and F. Granelli, "Soft SLICE: policy-based dynamic spectrum slicing in 5G cellular networks," in *IEEE International Conference on Communications*, pp. 1–6, Kansas City, MO, USA, 2018.
- [15] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Augusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.
- [16] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y. Liang, "Intelligent resource scheduling for 5G radio access network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7691–7703, 2019.
- [17] H. M. Soliman and A. Leon-Garcia, "QoS-aware frequency-space network slicing and admission control for virtual wireless networks," in *IEEE Global Communications Conference*, pp. 1–6, Washington, DC, USA, 2018.
- [18] Q. Ye, W. Zhuang, S. Zhang, A. Jin, and X. Shen, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9896–9910, 2018.
- [19] P. Zhao, H. Tian, S. Fan, and A. Paulraj, "Information prediction and dynamic programming-based RAN slicing for mobile edge computing," *IEEE Wireless Communications Letters*, vol. 22, no. 4, pp. 614–617, 2018.
- [20] Y. L. Lee, J. Loo, T. C. Chuah, and L. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, 2018.
- [21] Y. Xiao, M. Hirzallah, and M. Krunz, "Distributed resource allocation for network slicing over licensed and unlicensed bands," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2260–2274, 2018.
- [22] S. M. A. Kazmi, N. H. Tran, T. M. Ho, and C. S. Hong, "Hierarchical matching game for service selection and resource purchasing in wireless network virtualization," *IEEE Communications Letters*, vol. 22, no. 1, pp. 121–124, 2018.
- [23] P. Caballero, A. Banchs, and G. de Veciana, "Network slicing for guaranteed rate services: admission control and resource allocation games," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6419–6432, 2018.
- [24] S. D'Oro, F. Restuccia, T. Melodia, X. Costa-perez, and A. Azorra, "Low-complexity distributed radio access network slicing: algorithms and experimental results," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2815–2828, 2018.
- [25] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez, "Network slicing games: enabling customization in multi-tenant networks," *IEEE Conference on Computer Communications*, vol. 27, pp. 1–9, 2019.
- [26] Y. Jia, H. Tian, S. S. Fan, P. Zhao, and K. Zhao, "Bankruptcy game based resource allocation algorithm for 5G Cloud-RAN slicing," in *IEEE Wireless Communications and Networking Conference*, pp. 1–6, Barcelona, Spain, 2018.
- [27] J. Zheng, P. Caballero, G. de Veciana, S. J. Saek, and A. Banchs, "Statistical multiplexing and traffic shaping games for network slicing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2528–2541, 2018.
- [28] H. Halabian, "Distributed resource allocation optimization in 5G virtualized networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 627–642, 2019.
- [29] G. Zhao, S. Qin, G. Feng, and Y. Sun, "Network slice selection in softwarization based mobile networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, pp. 1–17, 2019.
- [30] S. Malla Reddy, B. Sergio, K. Wolfgang, and T. Srisakul, "Service-based slice selection function for 5G," in *IEEE Global Communications Conference*, pp. 1–6, Washington, DC, USA, 2016.
- [31] X. An, C. Zhou, R. Trivisonno, R. Guerzoni, and K. Alexandros, "On end to end network slicing for 5G communication systems," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, pp. 1–11, 2017.
- [32] V. K. Choyi, A. H. Ayman, and Y. Shah, "Network slice selection, assignment and routing within 5G networks," in *IEEE Conference on Standards for Communications and Network*, pp. 1–7, Berlin, Germany, 2016.

- [33] H. Xun, R. Chai, G. Jiang, and H. Li, "A joint utility optimization based virtual AP and network slice selection scheme for SDWNs," in *IEEE International Conference on Communications and Networking in China*, pp. 448–453, Shanghai, China, 2015.
- [34] J. Xu, S. Kim, and D. Byun, "Method and apparatus for supporting network slicing selection and authorization for new radio access technology," PCT/US2017/70331785A12017.
- [35] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings*, pp. 157–163, New Brunswick, NJ, 1994.
- [36] T. Börgers and R. Sarin, "Learning through reinforcement and replicator dynamics," *Journal of Economic Theory*, vol. 77, no. 1, pp. 1–14, 1997.
- [37] C. Qian, C. Luo, F. Hao, L. T. Yang, and G. Min, "Energy-efficient dynamic network selection in heterogeneous wireless networks," in *IEEE International Conference on Green Computing and Communications*, pp. 71–76, Beijing, China, 2013.
- [38] K. Tuyls, D. Heytens, and A. Nowe, *Extended Replicator Dynamics as a Key to Reinforcement Learning in Multi-agent Systems*, Springer, Berlin, 2003.
- [39] M. G. C. Resende and C. C. Ribeiro, *Greedy Randomized Adaptive Search Procedures: Advances and Extensions*, Springer, Berlin, 2018.