Opportunistic Admissibility and Resource Allocation For Slicing-based Radio Access Networks

Zihao Qi [1,2]
[1]*China Electric Power Research Institute*
[2] *Beijing University of Posts and Telecommunications*
Beijing, China
qizihao2021@163.com

Bin Cao*
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
caobin65@163.com

Long Zhang
National Key Laboratory of Science and Technology on Communication
University of Electronic Science and Technology of China
zhanglong3211@yeah.net

Wei Hong
Beijing Xiaomi Mobile Software
Beijing, China
hongwei@xiaomi.com

Zhongyuan Zhao
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
zyzhao@bupt.edu.cn

Jie Tong
China Electric Power Research Institute
Beijing, China
tongjie1@epri.sgcc.com.cn

Xiang Bai
China Electronic Technology Cyber Security Co., Ltd.
Beijing, China
13688042043@163.com

Network slicing (NS) is envisioned as a promising technology to meet the extremely diversified service requirements of users for future mobile networks. In radio access networks (RAN) slicing, the service providers (SPs) can rent network slicing instances from the infrastructure provider (InP) to meet the requirements of network services. However, both SPs and InP face the challenges of maintaining the quality of user experience and high profit in a dynamic environment, arising from random arrivals and departures of slice requests, uncertain resource availability, and multi-dimensional resource allocation. Therefore, admissibility and resource allocation become more complicated than that in traditional mobile networks. This paper proposes an opportunistic admissibility and resource allocation (OAR) policy to deal with the above challenges. To cope with the randomness of slice requests and resource availability, we first formulate this issue as a Markov Decision Process (MDP) problem to obtain the optimal admissibility strategy while maximizing the overall reward. Furthermore, we adopt a buyer-seller game-theoretic approach to optimize resource allocation, motivating SPs and InP to maximize their rewards. Numerical results show that the proposed OAR policy makes reasonable decisions effectively and steadily and outperforms the baseline scheme for system reward.

Keywords—resource allocation, Markov Decision Process，game

This work was supported in part by the National Key R\&D Program of China(2020YFB0905902), in part by the Fundamental Research Funds for the Central Universities(No.2020RC10), in part by the Beijing Natural Science Foundation (Grant L182039), in part by the National Natural Science Foundation of China (Grant 61971061), in part by the Sichuan International Science and Technology Innovation Cooperation/Hong Kong, Macao and Taiwan Science and Technology Innovation Cooperation Project(2019YFH0163), in part by the Key Research and Development Project of Sichuan Provincial Department of Science and Technology(2018JZ0071).

I. INTRODUCTION

Driven by the development of software-defined networking (SDN) and network function virtualization (NFV) technologies, network slicing is introduced in 5G to fulfill the ubiquitous delivery of delay-sensitive and computation-intensive network services. By decoupling network functions from proprietary hardware, the physical network is logically split into multiple isolated virtual networks, i.e., network slices. According to the different quality of service (QoS) and service level agreements (SLAs), network slicing can be dynamically customized by allocating resources and functions for specific network services.

Although various related work[1] has carefully investigated whether to accept slice requests or not and how to allocate resources, there remain other essential issues “whether”, “when” and “how” to admit slice requests in a dynamic environment. Especially in radio access networks (RANs), multi-dimensional and heterogeneous resources should also be sliced for provisioning tailored services, including radio, computing and storage resources. Assisted with network slicing, the infrastructure provider (InP) is responsible for splitting and maintaining network slices, and the service providers (SPs) can rent network slices from InP to meet their requirements.

From the perspective of SPs, heterogeneous resources leased from InP should be scheduled to match slice requests from different vertical industries. From the perspective of InP, due to the random arrivals and departures of slice requests and uncertain resource availability, InP might be lightly or heavily loaded over time. In the former case, “how” to admit is the main problem in network slicing. Nevertheless, “whether” to admit immediately should be answered first. If yes, multi-dimensional resources allocation should be performed in “how” to admit.

Otherwise, "when" to admit should be considered in future, or else failure happens.

In this paper, we design an opportunistic admissibility and resource allocation (OAR) policy to answer "whether", "when" and "how" to admit. Our major contributions are summarized as follows. (i) To better answer the questions of "whether", "when" and "how" to admit slicing requests, we design an opportunistic admissibility and resource allocation (OAR) policy. (ii) We formulate the original problem as a Markov Decision Process (MDP) problem and get the optimal admission policy to decide "whether" and "when" to admit slicing requests. (iii) To answer "how" to admit slicing requests, we reformulate the MDP problem as a buyer-seller game. Through this buyer-seller game, we can get the optimal resource allocation and price policy.

II. System Model and Function Descriptions

We consider a general slicing-based network model which consists of SPs, resource management and orchestration (RMO), and InP. SPs are responsible for requesting and leasing resources according to the arrival and departing slice requests. By implementing admission and negotiation of resource and price, RMO can deal with the three fundamental problems, i.e., whether a slice request is accepted, how much resources are allocated and how much price is paid for the accepted slice requests. Based on the accepted requests, InP rents networking, computing, and storage resources for SPs.

In the system, SPs can provide classes of slices containing different QoS, which can support diverse service types, such as industry slices and IoT slices. We denote the class of slice as $\mathcal{CS} = \{1, \dots, i, \dots, I\}$. Let $z \in \mathcal{RT}$ represent different resource types. $n_{i,z}(t)$ is the available number of resource units of z-th type for SP i in stage t. In the network slicing, we assume that there are $n_i(t)$ slice requests in stage t. Each slice request arrives and departs randomly. Our objective is to make an admissibility and resource allocation policy to satisfy the various requirements of slice requests while considering the utility, resource cost, and waiting cost simultaneously. Next, we will describe these functions in details.

A. Number of Slice Requests

First, the number of slice requests $n_i(t)$ in the t-th stage consists of newly arriving slice requests $n_i^a(t)$, waiting slice requests $n_i^w(t)$ and departing slice requests $n_i^d(t)$, expressed as

$$n_i(t) = n_i^a(t) + n_i^w(t) - n_i^d(t). \quad (1)$$

The number of waiting slice requests in stage t is denoted as

$$n_i^w(t) = n_i(t-1) - n_i^v(t-1), \quad (2)$$

where $n_i^v(t-1) = \sum_{j=1}^{n_i(t-1)} a_{i,j}(t-1)$, when $a_{i,j}(t-1) = 1$, it indicates that slice request j would be admitted in stage $t-1$. Otherwise, $a_{i,j}(t-1) = 0$, it would defer to the next stage. Furthermore, we assume that $n_i^a(t)$ follows an independent homogeneous poisson point process (HPPP), the probability mass function (PMF) of $n_i^a(t)$ is denoted as follows.

$$P_a(k) = P(n_i^a(t) = k) = \frac{[\lambda_i(t)T]^k}{k!} e^{-[\lambda_i(t)T]}, k = 0,1, \dots, \quad (3)$$

where $\lambda_i(t)$ is denoted as the average arrival rate of new slice requests in stage t, and T is the interval between contiguous stages. Similar to newly arriving slice requests, the number of departing slice requests is distributed as an HPPP with the average departure rate of $\mu_i(t)$. The PMF of departing slice requests is given as follows.

$$P_d(l) = P_d(n_i^d(t) = l) = \frac{[\mu_i(t)T]^l}{l!} e^{-[\mu_i(t)T]}, l = 0,1, \dots, \quad (4)$$

without loss of generality, the net number of slice requests h is equal to $h = k - l + w$, where w is the number of waiting slice requests. For simplicity, we consider that slice arrivals and departures are independent stochastic process in each stage, we can derive the probability of the net number of slice requests as

$$\begin{aligned} &P(h = k - l + w) \\ &= \sum_{m=-\infty}^{\infty} P_d(l = m) P_a(k = h + m - w) \\ &= e^{-[\mu_i(t)+\lambda_i(t)]T} \sum_{m=0}^{\infty} \frac{[\mu_i(t)T]^m [\lambda_i(t)T]^{(h+m-w)}}{m!\,(h+m-w)!} \end{aligned} \quad (5)$$

B. Utility Function

For SP i, slice request j rents $x_{i,j,z}(t)$ unit resource of type z to meet its QoS requirements. $\delta_{i,j,z}$ indicates how many resource units of type z should be rented for slice request j. To evaluate the satisfactory of slice request j, the logarithmic function, which has been widely used in previous works [2] is adopted as the utility function, give as $u\left(x_{i,j,z}(t)\right) = \log\left(1 + \delta_{i,j,z} x_{i,j,z}(t)\right)$ when $x_{i,j,z}(t) > 0$ and $u\left(x_{i,j,z}(t)\right) = -\infty$. In the following, we formulate the utility function of SP i for slice request j as

$$U_{i,j,z}(t) = a_{i,j}(t)\theta_{i,z} u\left(\delta_{i,j,z} x_{i,j,z}(t)\right). \quad (6)$$

where $\theta_{i,z}$ is the adjustment factor, and $\delta_{\mathrm{i,j,z}}$ is the required resource units to determine how many resource units of type \$z\$ should be rented for slice request j.

C. Resource Cost Function

Since available z-th type of resource units is limited and dynamic, the corresponding resource cost should be considered. We define the z-th resource unit cost $\beta_{i,z}(t)$ in stage t as follows.

$$\beta_{i,z}(t) = \pi_{i,z} + \varepsilon_{i,z}\tau_{i,z}. \quad (7)$$

where $\pi_{\mathrm{i,z}}$ is the resource unit cost priced by InP. $\varepsilon_{i,z}$ is the unit price of the z-th resource unit, $\tau_{i,z} = \frac{\sum_{j=1}^{n_i(t)} a_{i,j}(t)\delta_{i,j,z}}{n_{i,z}(t)}$. We can observe that if the network load is light, $\beta_{i,z}(t)$ would be low, which encourages SP to accept slice requests in the current stage. Otherwise, $\beta_{i,z}(t)$ becomes high, and SP i may refuse some slice requests in the current stage. Based on $\beta_{i,z}(t)$, we then can calculate the resource cost function $C_{i,j,z}^r(t)$ as follows.

$$C_{i,j,z}^r(t) = a_{i,j}(t)\delta_{i,j,z} x_{i,j,z}(t)\beta_{i,z}(t). \quad (8)$$

D. Waiting Cost Function

The waiting cost is mainly incurred by stage duration T and slice life cycle SLC from the current stage to the next stage. In stage t, the waiting cost function of slice request j for SP i is defined as follows.

$$C_{i,j}^{w}(t) = \left(1 - a_{i,j}(t)\right) Ceil\left(\frac{SLC}{T}\right) c_{i,j}^{w}. \quad (9)$$

where $c_{i,j}^{w}$ is the waiting unit cost of slice request j on SP i.

E. Immediate Reward Function

Based on the above description of utility and cost functions, the immediate reward function in stage t for slice request j can be calculated as the utility minus resource cost and waiting cost.

$$R_{i,j,z}(t) = U_{i,j,z}(t) - C_{i,j,z}^{r}(t) - C_{i,j}^{w}(t) \quad (10)$$

III. Opportunistic Admissibility and Resource Allocation Formulation

Due to the random arrival and departure of slice requests and limited resources, the admissibility and resource allocation policy would cause the unbalanced load cross SPs in the current stage while affecting resource cost and decision-making in the future stage. Therefore, admissibility and resource allocation policy should consider both the current and future stages simultaneously. To this end, we formulate this issue as a MDP optimization problem which is presented as a tuple $(\mathbb{S}, \mathbb{A}, \Pr, R)$ In the following section, the details will be shown separately.

A. System State and Action Space

The system states include the number of slice requests $n_i(t)$ and the available resource units $n_{i,z}(t)$. Thus, the system state space is given by $\mathbb{S}$. Specifically, one system state in stage t is $s_i(t) = \left\{\left(n_i(t), n_{i,z}(t)\right) \in \mathbb{S} | z \in \mathcal{RT}\right\}$. The action space is denoted as $\mathbb{A}$, and one action space in stage t is $a_i(t) = \left\{a_{i,j}(t) \in \mathbb{A} | j \in \{1, \dots, n_i(t)\}\right\}$, which is a set of admissibility actions for $n_i(t)$ slice requests.

B. System Transition Probability

In this part, we denoted the system transition probability as as $Pr\left(s_i(t+1) \middle| s_i(t), a_i(t)\right)$. Taking actions $a_i(t)$ in the current stage, the system would transfer from the current state $s_i(t)$ to another state $s_i(t+1)$ in the next stage with probability $Pr\left(s_i(t+1) \middle| s_i(t), a_i(t)\right)$.

1) Transition probability of $n_i(t)$

According to the arrived $n_i^a(t)$, accepted $n_i^v(t)$ and departed n_i^d slice requests, we can obtain the transition probability of slice requests as follows.

$$Pr\left(n_i(t+1) \middle| n_i(t), a_i(t)\right)$$
$$Pr\left(n_i(t+1) = n_i^a(t+1) + n_i^w(t+1) - n_i^d(t+1) \middle| n_i(t), a_i(t)\right)$$
$$Pr\left(n_i^a(t+1) - n_i^d(t+1) = n_i(t+1) - n_i(t) + n_i^v(t) \middle| n_i(t), a_i(t)\right) \quad (11)$$

2) Transition probability of $n_{i,z}(t)$

By observing resource cost in (7), the state transition of $n_{i,z}(t)$ depends on the available resource units and adopted actions. The available resource units $n_{i,z}(t)$ is affected by the random occupy and release of resource units of SP i. To represent this randomness, we consider that available resource units are determined by service rate v_i of SP i. According to queue theory, we can obtain an estimate of load rate $\rho_i(t) = \lambda_i(t)/v_i(t)$ in stage t, $\rho_i(t) < 1$, which indicates the rejection degree for slicing requests. On the other hand, available resource units $n_{i,z}(t)$ is also affected by the idle degree of z-th resource units, i.e., $n_{i,z}(t)/(n_{i,z}(t) + n_{i,z}^o(t))$. Based on the above analysis, we define a resource availability $\eta_{i,z}(t)$ of successfully leasing one resource unit z in stage t as follows.

$$\eta_{i,z}(t) = E_{\sim\lambda_i(t), v_i(t)}\left[\left(1 - \rho_i(t)\right)\right] \frac{n_{i,z}(t)}{n_{i,z}(t) + n_{i,z}^o(t)}. \quad (12)$$

Next, we discuss how to obtain the transition probability $P\left(n_{i,z}(t+1) \middle| n_{i,z}(t), a_i(t)\right)$. According to $a_i(t)$, the set of possible resource units from the current state to the next state is expressed as $n_{i,z}(t+1) = \left\{n_{i,z}(t), n_{i,z}(t) - 1, \cdots, n_{i,z}(t) - \zeta\right\}$, where $\zeta = min\left\{\sum_{j=1}^{n_i(t)} a_{i,j}(t)\delta_{i,j,z}, n_{i,z}(t)\right\}$. To analyze the generic transition probability $P\left(n_{i,z}(t+1) \mid n_{i,z}(t), a_i(t)\right)$, let $\mathbb{R} = \left\{\mathbb{R}_1, \mathbb{R}_2, \dots, \mathbb{R}_l, \dots, \mathbb{R}_{2^{n_i(t)}}\right\}$ be the set of all the possible combinations of actions $a_i(t)$, where $\mathbb{R}_l$ is the l-th combination of actions $a_i(t)$. In addition, we use $|\mathbb{R}_l|$ to represent the required resource units for the l-th combination. Then, $\Pr\left(n_{i,z}(t+1) \mid n_{i,z}(t), a_i(t)\right)$ from state $n_{i,z}(t)$ to state $n_{i,z}(t+1)$ can be expressed in (13).

$$\begin{aligned} &\Pr\left(n_{i,z}(t) - \zeta \mid n_{i,z}(t), a_i(t)\right) \\ &= \prod_{j=1}^{n_i(t)} \left(1 - a_{i,j}(t)\right)\left(1 - \eta_{i,z}(t)^{\delta_{i,j,z}}\right) \times \\ &\sum_{\mathbb{R}_l \in \mathbb{R}, |\mathbb{R}_l| = \zeta} \left(\prod_{\forall k \in \mathbb{R}_l, a_{i,k}(t) = 1} \frac{a_{i,k}(t)\eta_{i,z}(t)^{\delta_{i,j,z}}}{\left(1 - a_{i,k}(t)\right)\left(1 - \eta_{i,z}(t)^{\delta_{i,j,z}}\right)}\right) \end{aligned} \quad (13)$$

3) System transition probability

Based on the above analysis, system state space of $s_i(t)$ can be regarded as a Cartesian product of $n_i(t)$ and $n_{i,z}(t)$. The system transition probability $\Pr(s_i(t+1) \mid s_i(t), a_i(t))$ can be calculated as

$$\begin{aligned} &\Pr(s_i(t+1) \mid s_i(t), a_i(t)) \\ &= \Pr(n_i(t+1) \mid n_i(t), a_i(t)) \times \\ &\left\{\Pr\left(n_{i,z}(t+1) \mid n_{i,z}(t), a_i(t)\right), z \in \mathcal{RT}\right\} \end{aligned} \quad (14)$$

C. Optimal Admissibility and Resource Allocation Policy in MDP Model

In this paper, we formulate the admissibility and resource allocation policy as a MDP problem. In the MDP model, the policy is denoted as $\Phi(s_i(t), a_i(t))$, indicating to take the action $a_i(t)$ in state $s(t)$. The optimal admissibility policy aims to optimize admission actions to decide whether to be accepted by RMO or wait according to the value function. The value function $R(s_i(t), a_i(t))$ is defined as follows

$$R(s_i(t), a_i(t)) = \sum_{i=1}^{I} \sum_{j=1}^{n_i(t)} \sum_{z \in \mathcal{RT}} \Big[R_{i,j,z}(t) + \gamma \sum_{s_i(t+1) \in \mathbb{S}} \Pr(s_i(t+1) \mid s_i(t), a_i(t)) V(s_i(t+1)) \Big]. \quad (15)$$

$$\text{s.t. } C_1: x_{i,z}^{\min} \le x_{i,j,z}(t) \le x_{i,z}^{\max}$$

$$C_2: \sum_{i=1}^{I} \sum_{j=1}^{n_i(t)} a_{i,j}(t) \delta_{i,j,z} x_{i,j,z}(t) \le x_z^{\max}, z \in \mathcal{RT}$$

$V(s_i(t+1)) = \max_{\Phi(s_1(t+1), a_1(t+1))} R(s_i(t+1), a_i(t+1))$ is represented as the optimal value function of state $s_i(t)$ to evaluate the reward with different $a_i(t)$. $\gamma \sum_{s_1(t+1) \in \mathbb{S}} P(s_i(t+1) \mid s_i(t), a_i(t)) V(s_i(t+1))$ is denoted as the expected sum of reward functions in the future stage associated with γ.

IV. Buyer-Seller Game Based Strategy Analysis

To get resource allocation policy, we reformulate this MDP optimization problem as a buyer-seller game to maximize the individual reward to make resource allocation and price policy.

A. Buyer Level

We define SP i as the i-th buyer (b_i), and InP is defined as the seller s participating in the game. To satisfy various QoS requirements, b_i needs to rent a certain amount of resources from s_i. Meanwhile, b_i needs to pay an amount of payment to s. Due to rationality and selfishness, b_i aims to maximize its reward as much as possible. Therefore, the optimization objective of b_1 is denoted as follows, $R(s_i(t), a_i(t))$ can be got from (15).

$$\begin{aligned} U_{b_i} &= R(s_i(t), a_i(t)) \\ \text{s.t.} \quad & C_1: x_{i,z}^{\min} \le x_{i,j,z}(t) \le x_{i,z}^{\max} \end{aligned} \quad (16)$$

B. seller Level

Due to rationality and selfishness, s also aims to maximize its reward by providing resources to b_i with competitive selling price $\pi_{i,z}(t)$, while minimizing the resource cost as little as possible. Since s provides resources to multiple buyers, the total reward is denoted as follow

$$\begin{aligned} U_s &= \sum_{i=1}^{I} \sum_{z \in \mathcal{RT}} (\pi_{i,z}(t) - \beta_{i,z}^{o}) \sum_{j=1}^{n_i(t)} \delta_{i,j,z} x_{i,j,z}(t), \\ \text{s.t.} \quad & C_2: \sum_{i=1}^{I} \sum_{j=1}^{n_i(t)} a_{i,j}(t) \delta_{i,j,z} x_{i,j,z}(t) \le x_z^{\max}, z \in \mathcal{RT} \end{aligned} \quad (17)$$

C. Optimal Solution

In this section, we analyze the optimal solution to determine the selling price and resource allocation. The reward for b_i is shown in (16), we can derive $\frac{\partial^2 U_{b_i(t)}}{(\partial x_{i,j,z}(t))^2} < 0$. Therefore, $U_{b_i}(t)$ is a strictly concave function of $x_{i,j,z}(t)$. Letting $\frac{\partial U_{b_i}(t)}{\partial x_{i,j,z}(t)} = 0$, we can get

$$x'_{i,j,z}(t) = \frac{1}{\delta_{i,j,z}} \left(\frac{\theta_{i,z}}{(\pi_{i,z} + \varepsilon_{i,z} \tau_{i,z}) \ln 2} - 1 \right). \quad (18)$$

Accordingly, the optimal solution of b_i is obtained as follows.

$$x_{i,j,z}^{*}(t) = \begin{cases} x_{i,z}^{\min}, & \text{if } x'_{i,j,z}(t) \le x_{i,z}^{\min} \\ x'_{i,j,z}(t), & \text{if } x_{i,z}^{\min} \le x'_{i,j,z}(t) < x_{i,z}^{\max} \\ x_{i,z}^{\max}, & \text{if } x'_{i,j,z}(t) \ge x_{i,z}^{\max} \end{cases} \quad (19)$$

In (18) and (19), we can see that resource unit $x_{i,j,z}(t)$ is related to $\pi_{i,z}(t)$, which is determined by seller s. Similarly, from (17), we can get the second-order derivative of U_s can be derived as $\frac{\partial^2 U_s}{(\partial \pi_{i,z}(t))^2} < 0$. Obviously, U_s is a concave function of $\pi_{i,z}(t)$. Meanwhile, since the constraint C_2 is affine, the optimal solution of seller s can be calculated using Lagrangian multiplier ν_z as follows,

$$\begin{aligned} L_s &= \sum_{i=1}^{I} \sum_{z \in \mathcal{RT}} \left(\pi_{i,z}(t) - \beta_{i,z}^{o} \right) \sum_{j=1}^{n_i(t)} \delta_{i,j,z} x_{i,j,z}(t) - \\ & \sum_{z \in \mathcal{RT}} \nu_z \sum_{i=1}^{I} \sum_{j=1}^{n_i(t)} \left(a_{i,j}(t) \delta_{i,j,z} x_{i,j,z}(t) - x_z^{\max} \right), \\ \text{s.t.} \quad & C_3: \pi_{i,z}(t) \ge 0 \end{aligned} \quad (20)$$

using KKT conditions, we can get the optimal selling price $\pi_{i,z}^{*}(t)$ is denoted as

$$\pi_{i,z}^{*}(t) = \beta_{i,z}^{o} + \frac{\nu_z \sum_{j=1}^{n_i(t)} \delta_{i,j,z} \frac{\partial x_{i,j,z}}{\partial \pi_{i,z}(t)} - \sum_{j=1}^{n_i(t)} \delta_{i,j,z} x_{i,j,z}(t)}{\sum_{j=1}^{n_i(t)} \delta_{i,j,z} \frac{\partial x_{i,j,z}(t)}{\partial \pi_{i,z}(t)}}, \quad (21)$$

where ν_z can be updated by using classical subgradient method.

D. Opportunistic Admissibility and Resource Allocation Policy

Nest, we propose the OAR policy to solve the opportunistic admissibility and resource allocation problem. OAR policy is shown in $Algorithm$ 1. First, we obtain the optimal solution of admissibility $\Phi(s_i(t), a_i(t))$ based on MDP model. Then, we obtain the optimal resource allocation based on the game. The buyer-seller game begins to execute negotiation process, such as exchanging negotiation information [2] $\left(x_{i,j,z}(t), \frac{\partial x_{i,j,z}(t)}{\partial \pi_{i,z}(t)} \right)$ from the buyer to the seller, and $\pi_{i,z}(t)$ from the seller to the buyer. With information exchanging, the buyer and the seller update their own reward function and iterate until SE solutions $x_{i,j,z}^{SE}(t)$ and $\pi_{i,z}^{SE}(t)$ are obtained.

Algorithm 1: Opportunistic Admissibility and Resource Allocation Policy

Input : $n_i^a(t), n_i^a(t), n_i^{\omega}(t), \delta_{i,j,z}, \lambda_i(t), \mu_i(t), \theta_{i,z}$
Output: $x_{i,j,z}^{SE}(t), \pi_{i,z}^{SE}(t)$
Set the initial stage $t = 0$
For all $s_i(t) \in \mathbb{S}$ do
  Compute $V(s_i(t+1)) = \max 1_{\Phi(s_i(t), a_i(t))} R(s_i(t), a_i(t))$
  If $max_{s_i(t) \in \mathbb{S}} \mid V(s_i(t+1)) - V(s_i(t)) \mid < threshold$ Then
    Output $\Phi(s_i(t), a_i(t))$
    Break
  else
    $V(s_i(t+1)) = V(s_i(t))$

End if
End For
Set $converged \leftarrow false$, number index $iter \leftarrow 0$
$x_{i,j,z}(t) = x_{i,j,z}^{init}$, $\pi_{i,z}(t) = \pi_{i,z}^{init}$
While not converged in current stage
  $iter = iter + 1$
  For each buyer i = 1:I do
    Update $x_{i,j,z}^{*}(t)$
    If $\frac{\partial L_s}{\partial \pi_{i,z}(t)} > 0$ then
      Update $\pi_{i,j,z}^{*}(t)$
    else
      $converged \leftarrow true$
      $x_{i,j,z}^{SE}(t) = x_{i,j,z}(t)$
      $\pi_{i,z}^{SE}(t) = \pi_{i,t}(t)$
    End if
  End for
End while

V. Performance Evaluation and Discussion

In this section, we design experiments to evaluate the effectiveness and performance of the proposed OAR policy. To validate the performance of the proposed policy, we introduce the classical Greedy scheme, and Max-Min Fairness (MMF) scheme as the benchmark.

A. Parameter Settings

Considering a resource-constrained slicing system, we set the maximum resource capacity of networking, storage, and computing resources as 2bps, 20GB, and 20 CPUs, respectively. According to [3], $\delta_{i,j,z}$ is set as [1,1,2], and $x_{i,j,z}^{init}(t)$ is set as 200Mbps, 2GB, 1 CPU. According to [4], SLC of slice request is set as 35secs for eMBB slices. In addition, $\varepsilon_{i,z}$ is set as 0.45, 0.2 and 0.4 for z-th resource type, $\gamma = 0.9$, $\lambda_i = 8$, $\mu_i = 2$, $v_i(t) = 10$, $\theta = 3.5$, $T = 5$,$c_{i,j}^{w} = 0.1$.

B. Performance Analysis

The proposed OAR scheme composes of two stages. In the first stage, OAR iteratively updates the value function for each state until convergence is achieved. According to the optimal action in first stage, buyer-seller game iteration is executed to obtain the optimal resource and price in the second stage.

Fig. 1 shows the iteration process of the value function for a concrete state. We can see that the value function eventually converges to a stable value after about 15 iterations, which validates the effectiveness of the OAR scheme in the first stage.

From Fig. 2 and Fig. 3, we can observe that the game process of resource and selling price can reach equilibrium after about eight iterations.

Fig. 4 shows the performance in term of reward by varying the number of slice requests. We can see that the reward of the OAR scheme outperforms compared with that of other baseline schemes. That is because the proposed OAR scheme considers the impact of slice requests and available resources on decision-making and makes a more reasonable admission and resource allocation policy. In contrast, the greedy scheme only satisfies the maximum resource requirements for the partial slice requests, hindering the admission for more slice requests. MMF scheme admits all slice requests to be accepted by SPs, while the accepted slice requests obtain fewer resources. Therefore, the proposed OAR scheme can make a trade-off between slicing numbers and resource allocation.

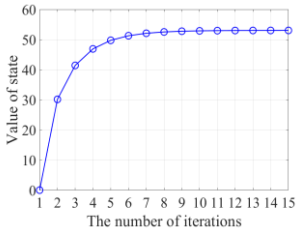

Fig. 1. Iteration process of value

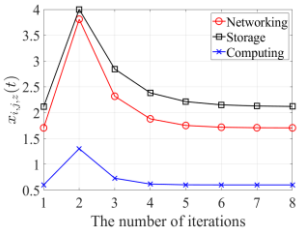

Fig. 2. Iteration processof resource.

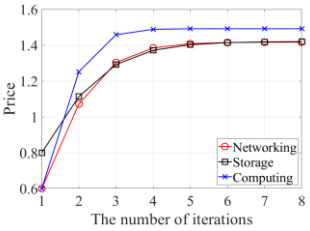

Fig. 3. Iteration process of price

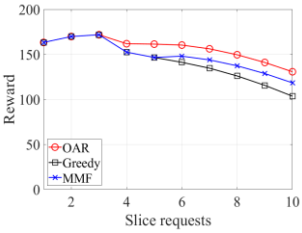

Fig. 4. Reward comparisons

VI. Conclusion

In this paper, we investigate the issue of "whether" and "when" to admit slicing requests and "how" to allocate multi-dimensional and heterogeneous resource types. Considering random arrivals and departures of slicing requests, we design an opportunistic admissibility and resource allocation (OAR) strategy to answer the above issue. To get the optimal admission policy, we formulate the optimization problem as a Markov Decision Process problem to maximize system reward. Then, a buyer-seller game is introduced to get the optimal resource allocation and price between SPs and InP. Simulation results show that the proposed OAR strategy makes reasonable decisions and outperforms the baseline schemes.

References

[1] M. R. Rahman and R. Boutaba, "SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization," in IEEE Transactions on Network and Service Management, vol. 10, no. 2, pp. 105-118, June 2013

[2] C. Tham and B. Cao, "Stochastic Programming Methods for Workload Assignment in an Ad Hoc Mobile Cloud," in IEEE Transactions on Mobile Computing, vol. 17, no. 7, pp. 1709-1722, 1 July 2018

[3] N. Van Huynh, D. T. Hoang, D. N. Nguyen and E. Dutkiewicz, "Real-Time Network Slicing with Uncertain Demand: A Deep Learning Approach," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1-6

[4] S. Vittal, M. K. Singh and A. Antony Franklin, "Adaptive Network Slicing with Multi-Site Deployment in 5G Core Networks," 2020 6th IEEE Conference on Network Softwarization (NetSoft), 2020, pp. 227-231

Please help us understand your paper better by completing below form, and it will not be published

1st Author	**Position:** Mr.
	Research Field: mobile edge computing
	Homepage URL:
2nd Author	**Position:**
	Research Field:
	Homepage URL:
3rd Author	**Position:**
	Research Field:
	Homepage URL:
4th Author	**Position:**
	Research Field:
	Homepage URL:
5th Author	**Position:**
	Research Field:
	Homepage URL:
6th Author	**Position:**
	Research Field:
	Homepage URL:
Add more rows if necessary!	