TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
Trans. Emerging Tel. Tech. ; **00**:1–12
DOI: 10.1002/ett

RESEARCH ARTICLE

Network Slice Selection in Softwarization based Mobile Networks

Guanqun Zhao[1], Shuang Qin[1], Gang Feng*[1,2]

[1]National Key Laboratory of Communications, [2]Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, China.

ABSTRACT

Recently network slicing has been introduced as a key enabler to accommodate diversified services in NFV-enabled software-defined mobile networks. Although there has been some research work on network slice deployment and configuration, how user equipments (UEs) select the most appropriate network slice is still an essential yet challenging issue, as slice selection may substantially affect the resource utilization and user quality of service. In this paper, we investigate the optimal selection of end-to-end slices with aim of improving network resources utilization while guaranteeing the quality of service (QoS) of users. We formulate the optimal slice selection problem as maximizing the users satisfaction degree, and prove it is NP-hard. We thus resort to genetic algorithm (GA) to find a sub-optimal solution, and develop a GA based heuristic algorithm. The effectiveness of our proposed NS selection algorithm is validated via simulation experiments. Copyright © John Wiley & Sons, Ltd.

***Correspondence**

National Key Laboratory of Communications, University of Electronic Science and Technology of China, Chengdu 611731, China.

E-mail: fenggang@uestc.edu.cn

1. INTRODUCTION

Network slicing is based on the concept of network softwarization, i.e. novel technologies including network functions virtualization (NFV) and software-defined networking (SDN) [1-4]. An end-to-end network slice carries a set of flows belonging to various end users and has complete control over a collection of virtual resources. Multiple slices are mapped to the same underlying physical network.

In slice-based mobile networks, users need to select appropriate network slice according to the service requirements. A base station (BS) may be associated with multiple network slices. Each slice is allocated a certain amount of resources, and resource isolation between slices is enforced. The same type of network slice may cover multiple different base stations [4-5]. Therefore, the available resources and service load of different slices on the same BS are independent from each other, and the available resources and service load of the same type of slices on different base stations may vary. When a mobile user accesses the network, it needs to consider the available service capabilities and available resources of the network slice according to the service requirements and select an appropriate one to support the user applications. On the other hand, it is also necessary to consider the wireless channel conditions of the base station when determining a slice to access. Therefore, in slice-based mobile communication networks, access to a slice needs to consider not only the slice service as well as available resources, but also the radio channel conditions from the user to the corresponding BS of the slice.

In traditional UE association problem, only optimal matching between mobile users and access points is

Prepared using ettauth.cls [Version: 2012/06/19 v2.10]

considered. In comparison, in the slice selection problem, optimal matching between mobile users, network slices, and access points should be addressed. The design objective of traditional access point selection is usually network resource utilization maximization subject to radio channel conditions of access points and system capacity [8][15][16]. Individual user's service demands are ignored, and thus it is difficult to ensure the user's quality of service. In contrast, in slice-based mobile communication networks, user service demands should be carefully considered when performing slice selection or slice association. In the meantime, it is essential to maximize network resource utilization. Furthermore, in an end-to-end network slice architecture, it is also necessary to consider the constraints of the core network (CN) segment. Therefore, existing access point selection mechanisms and algorithms cannot be directly applied to slice selection problem.

Recently, end-to-end network slicing for mobile network has attracted a lot of research interest. While existing research mainly focuses on network architecture and feasibility analysis, little research effort is devoted to access selection mechanisms for network slicing [3-7]. The authors of [3] introduce the design principle of network slicing and investigate the deployment and implementation of the network function of slicing selection. In [4], a wireless network architecture based on end-to-end network slicing is proposed, and the basic procedure of slice selection is analyzed. The authors of [5] propose a network slice access scheme based on QoS Class Identifiers. The authors of [7] mainly focus on the feasible framework and basic procedure of slice selection. In [6], the authors address the problem of user's optimal access selection under the network architecture of virtual access points (VAP) and core network slices. However, only the deployment of network slicing in the core network is considered without considering the access network of a slice and providing services for users through end-to-end network slices. Therefore, research on selection of end-to-end network slice according to user's service requirements is still not adequate. It is imperative to investigate the selection of access points and slices according to the service requirements and the constraints of the network, and achieve optimal matching between users, BSs, and slices.

In this paper, we investigate wireless access selection mechanism for end-to-end network slicing from the perspective of optimal matching among users, BSs, and slices. We aim to maximize Satisfaction Degree (SD) in the system and formulate an optimization problem of end-to-end slice selection, which is proven NP-hard. We thus resort to genetic algorithm (GA) to solve the optimization problem. We conduct simulation experiments to validate the effectiveness of our proposed slice selection scheme. Numerical results demonstrate that our proposed GA based slice selection algorithm outperforms conventional received signal strength (RSS) based access point selection algorithm and greedy algorithm. It can also improve network resource utilization while guaranteeing the users' quality of service.

The rest of the paper is organized as follows. In Section II, we describe the system model and introduce the concept of Satisfaction Degree. Section III formulates the slice selection problem and proves its NP hardness. We present the solution to the optimal slice selection problem in Section IV. Section V provides numerical results for the performance comparison. Finally, Section VI concludes the paper.

2. SYSTEM MODEL

2.1. Network model

We consider a slice based mobile network model as shown in Fig. 1, which consists of the core network and access networks. There are M base stations and L users in the network. The Access and Mobility Management Function (AMF) of the core network is responsible for the deployment and management of network slices [9]. When a slice covers a BS, the site communicates with the AMF that manages the slice, for exchanging information about radio channel conditions, available network resources, and other necessary network states. AMF can use this information to perform slice selection and the associated resource allocation.

Let there be N network slices deployed based on the substrate network. A slice may cover multiple BSs, and corresponding transmission resources of the BSs are allocated to individual slices. Thus, a BS may also be used by multiple slices and provide access services for UEs with various demands. Let $B_{j,k}$ and $P_{j,k}$ denote the available

Trans. Emerging Tel. Tech. ; **00**:1–12 © John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

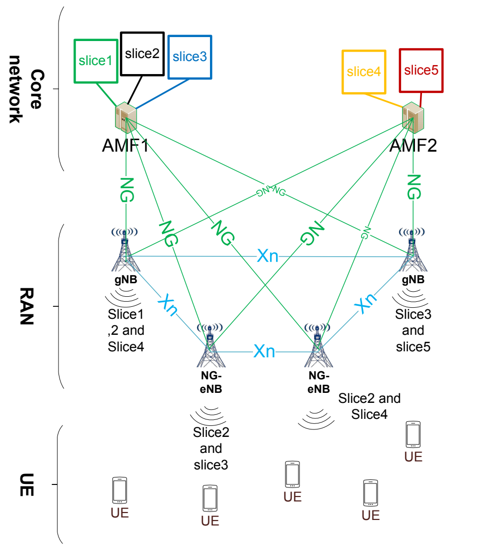

Figure 1. Network slice-based network architecture

transmission bandwidth and transmission power allocated to slice j at BS k respectively. We have $\sum_{j=1}^{N} B_{j,k} \leq B_k$ and $\sum_{j=1}^{N} P_{j,k} \leq P_k$, where B_k and P_k are the maximum transmission bandwidth and transmission power available at BS k.

2.2. Slice selection

Thus far, the basic procedure of UE association with a slice has been described in 3GPP documents [9][21]. However, the concrete slice selection scheme is not specified. In slice-based mobile networks, Slice Selection Function (SSF) handles the UE's initial Attach Request and New Session establishment request by selecting an appropriate slice for the UE based on the UE's subscription information, UE usage type, service type and UE capabilities. When a UE is attached to the network for the first time and has no valid slice ID, the RAN forwards the request to the AMF, which selects an appropriate slice for the UE based on assistance information provided by the UE. A Slice ID is represented by an NSSAI (Network Slice Selection Assistance Information) or S-NSSAI. As defined in TR 23.799 [21], NSSAI includes one or more S-NSSAIs (Single NSSAI). Each network slice is uniquely identified by a S-NSSAI.

When multiple UEs send access requests, because the network resources are limited, in order to satisfy the user's QoS requirements while optimizing resource utilization, the system needs to make a reasonable access decision based on the UE's availability information. The assistance information of the UE should include the current UE channel states, QoS requirement and other relevant information. The network selects an appropriate slice which includes the RAN slice part and the CN slice part for the UE based on the UE specified information and CN specified information. Then the UE is assigned a Slice ID, to support the selection of an AMF. If available, NG-RAN uses the information for routing the NAS message to the appropriate AMF which supports UE requested slices. From the procedure, we can see that slice selection is one of the key issues which affects both the network performance and UE QoS.

2.3. Satisfaction Degree

In this paper we use Satisfaction Degree (SD) as the optimization objective for slice selection, similar to that in [10]. Our design target is to improve the system resource utilization while guaranteeing the Quality of Service (QoS) of users, through solving the SD optimization problem.

Without loss of generality, let user i generate one service flow with required service rate $R_{req,i}$. Therefore, in the following, user i and flow i are equivalent. There are L service flows in the network which need to be carried by individual network slices. Suppose user i is associated with slice j on BS k, obtaining actual service rate $R_{i,j}^k$ from this slice. Based on Sigmoid function, we can define the utility function [11]

$$U_{i,j}^k = \frac{[\rho(R_{i,j}^k/R_{req,i})]^{\xi}}{1+[\rho(R_{i,j}^k/R_{req,i})]^{\xi}} \tag{1}$$

where ρ and ξ in (1) are constant greater than zero. Obviously, the value of $U_{i,j}^k$ increases with $R_{i,j}^k$. When the value of $R_{i,j}^k$ is much greater than $R_{req,i}$, $U_{i,j}^k$ gradually approaches 1. Therefore, if $U_{i,j}^k$ is directly used as the objective function of slice selection problem, it cannot reflect the performance change of the network when $R_{i,j}^k$ is much larger than $R_{req,i}$, which may cause excessive service for some users and unreasonable allocation of network resources. Similar to the idea of [10][18], we define the Satisfaction Degree (SD) function of user i connected to slice j through BS k as

$$f_{i,j}^k = (1 - e^{-\frac{U_{i,j}^k}{\rho(R_{i,j}^k/R_{req,i})}})/\sigma \tag{2}$$

where σ is the normalization factor used to ensure the range of the SD is [0,1], and σ can be expressed as

$$\sigma = 1 - e^{-\frac{1}{(\xi-1)^{1/\xi}+(\xi-1)^{(1-\xi)/\xi}}} \tag{3}$$

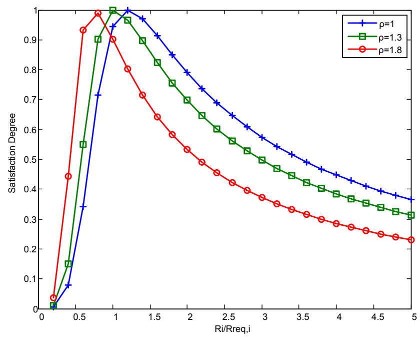

Figure 2. Satisfaction Degree vs $R_{i,j}^k/R_{req,i}$

According to (2), user's SD as a function of the service rate is shown in Fig.2. We can see that SD is a convex function of $R_{i,j}^k/R_{req,i}$, and there exist a maximum value 1 for SD. Parameter ρ determines the value of $R_{i,j}^k/R_{req,i}$ corresponding to the maximum SD. As shown in Fig. 2, when ρ=1.3 and $R_{i,j}^k/R_{req,i} = 1$, SD = 1. Therefore, in this paper, we consider ρ=1.3, and obtain the maximum SD when the service rate obtained by the user is equal to the demand rate. When $R_{i,j}^k < R_{req,i}$, the service rate obtained by the user does not meet the demand, the SD declines; and when $R_{i,j}^k > R_{req,i}$, The transmission resources allocated to users by the system exceed their requirements, which may cause the reduction of resource utilization, resulting in a decrease of SD.

3. PROBLEM FORMULATION

3.1. Problem Formulation

As described in Section 2, we consider that L users access the mobile communication network composed of M BSs and N slices. A service flow is only allowed to access one network slice through one BS to obtain the corresponding transmission service. Therefore, we can define variable $\alpha_{i,j}^k \in \{0,1\}$. $\alpha_{i,j}^k = 1$ if user *i* accesses slice *j* through BS *k* and $\alpha_{i,j}^k = 0$ otherwise. Let $b_{i,j}^k$ indicate the transmission bandwidth obtained by user *i* when it accesses slice *j* via BS *k*. Therefore, the service rate obtained by the user can be expressed as

$$R_{i,j}^k = b_{i,j}^k \log_2(1 + SINR_{i,j}^k) \tag{4}$$

where $\sum_{i=1}^{L} \alpha_{i,j}^k b_{i,j}^k \le B_{j,k}$, $SINR_{i,j}^k$ is the signal to noise ratio of users. Let G_i^k be the wireless channel gain of user *i* on BS *k*, $p_{i,j}^k$ be the transmission power assigned to user *i* by network slice *j* on AP *k*. The value of the $SINR_{i,j}^k$ can be expressed as

$$SINR_{i,j}^k = \frac{G_i^k \cdot p_{i,j}^k}{I_i^k + N_0} \tag{5}$$

where N_0 is the additive Gaussian white noise and $\sum_{i=1}^{L} \alpha_{i,j}^k p_{i,j}^k \le P_{j,k}$, $I_i^k = \sum_{k' \in M-k} \sum_{i' \in L-i} \sum_{j \in N} G_i^{k'} \cdot p_{i',j}^{k'}$ is the interference generated by other BSs when user *i* accesses BS *k*.

It should be noted that an end-to-end network slice should cover both the access and core networks of the mobile communication system. We assume that the total capacity of each network slice at core network is limited to C_j, and we thus have

$$\sum\nolimits_{k=1}^{M} \sum\nolimits_{i=1}^{L} \alpha_{i,j}^k R_{i,j}^k \le C_j \tag{6}$$

According to (2) and (4), we can obtain the SD of user *i* when it accesses slice *j* on BS *k*. In this paper, in order to maximize the utilization of system resources while satisfying the service demands of users, we consider the overall users' satisfaction Degree of the system as the design objective, and we can thus formulate the optimal slice selection problem as P1.

In problem P1, constraints (7-2) and (7-3) ensure that the total bandwidth and power allocated to UEs by NS *j* via BS *k* does not exceed the amount of available resources of NS *j* deployed on BS *k*. Constraint (7-4) indicates that the total amount of traffic for the slice service cannot exceed its capacity. Constraints (7-5) and (7-6) indicate that the transmission resources allocated to all slices at each BS cannot exceed the available resources of the BS, while constraint (7-7) indicates that each UE can only access to one NS via one BS as a time. According to the definition of variable $\alpha_{i,j}^k \in \{0,1\}$, P1 is a complex 0-1 integer

Trans. Emerging Tel. Tech. ; **00**:1–12 © John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

programming problem.

$$P1: \quad \max \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=1}^{M} a_{i,j}^{k} f_{i,j}^{k} \quad (7\text{-}1)$$

$$s.t. \sum_{i=1}^{L} a_{i,j}^{k} b_{i,j}^{k} \leq B_{j,k} \quad (7\text{-}2)$$

$$\sum_{i=1}^{L} a_{i,j}^{k} p_{i,j}^{k} \leq P_{j,k} \quad (7\text{-}3)$$

$$\sum_{k=1}^{M}\sum_{i=1}^{L} a_{i,j}^{k} R_{i,j}^{k} \leq C_j \quad (7\text{-}4)$$

$$\sum_{j=1}^{N} B_{j,k} \leq B_k \quad (7\text{-}5)$$

$$\sum_{j=1}^{N} P_{j,k} \leq P_k \quad (7\text{-}6)$$

$$\sum_{k=1}^{M}\sum_{j=1}^{N} a_{i,j}^{k} = 1 \quad (7\text{-}7)$$

3.2. Problem Complexity

For proving the NP-hardness of P1, we consider a special case, in which we assume that a slice in the network can only be associated with one BS. In this case, the slice is selected to determine the corresponding BS. Therefore, the variable $\alpha_{i,j}^{k}$ can be expressed as

$$\alpha_{i,j}^{k} = \alpha_{i,j}\alpha_{j,k} \quad (8)$$

where $\alpha_{i,j} \in \{0,1\}$, $\alpha_{i,j}$=1 indicates that user *i* accesses slice *j* and $\alpha_{i,j}$= 0 otherwise. Obviously, we have $\sum_{j=1}^{N}\alpha_{i,j} = 1$. Similarly, we define $\alpha_{j,k} \in \{0,1\}$. $\alpha_{j,k}$=1 indicates that slice *j* is associated with BS *k* and $\alpha_{i,j}^{k} = 0$ otherwise. In general, after NS is deployed, the association of its BS will not be changed in a short time. Therefore, $\alpha_{j,k}$ in (8) is a constant and $\sum_{k=1}^{M}\alpha_{j,k} = 1$.

Using (8), we can rewrite (7-1) as

$$\begin{aligned} &\sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=1}^{M} \alpha_{i,j}^{k} f_{i,j}^{k} \\ &= \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=1}^{M} \alpha_{i,j}\alpha_{j,k} f_{i,j}^{k} \\ &= \sum_{i=1}^{L}\sum_{j=1}^{N} \alpha_{i,j} \sum_{k=1}^{M} \alpha_{j,k} f_{i,j}^{k} \\ &= \sum_{i=1}^{L}\sum_{j=1}^{N} \alpha_{i,j} g_{i,j} \end{aligned} \quad (9)$$

where $g_{i,j}$ indicates the SD of user *i* accessing slice *j* and $g_{i,j} = \sum_{k=1}^{M}\alpha_{j,k} f_{i,j}^{k}$. We can also rewrite (7-4) as

$$\begin{aligned} &\sum_{i=1}^{L}\sum_{k=1}^{M} \alpha_{i,j}^{k} R_{i,j}^{k} \\ &= \sum_{i=1}^{L}\sum_{k=1}^{M} \alpha_{i,j}\alpha_{j,k} R_{i,j}^{k} \\ &= \sum_{i=1}^{L} \alpha_{i,j} \sum_{k=1}^{M} \alpha_{j,k} R_{i,j}^{k} \\ &= \sum_{i=1}^{L} \alpha_{i,j} r_{i,j} \leq C_j \end{aligned} \quad (10)$$

where $r_{i,j} = \sum_{k=1}^{M}\alpha_{j,k} R_{i,j}^{k}$ denotes the service rate obtained by user *i* accessing slice *j*.

We note that a slice can only be associated with one BS, and the available bandwidth and power resources are assigned to the corresponding slice by the BS when the slice is deployed and remain unchanged. Therefore, (7-2) and (7-3) can be simplified as

$$\sum_{i=1}^{L}\alpha_{i,j} b_{i,j} \leq B_j, \sum_{i=1}^{L}\alpha_{i,j} p_{i,j} \leq P_j \quad (11)$$

where B_j and P_j are the available transmission bandwidth and power resources of slice j. $b_{i,j}$ and $p_{i,j}$ are the transmission bandwidth and power obtained by user *i* accessing slice *j*.

Considering that the inequalities (7-5) and (7-6) are constant on both sides, we can ignore them. From (8)-(11), we can simplify problem P1 as

$$P2: \quad \max \sum_{i=1}^{L}\sum_{j=1}^{N} \alpha_{i,j} g_{i,j} \quad (12\text{-}1)$$

$$s.t. \sum_{i=1}^{L} \alpha_{i,j} r_{i,j} \leq C_j \quad (12\text{-}2)$$

$$\sum_{i=1}^{L} \alpha_{i,j} b_{i,j} \leq B_j \quad (12\text{-}3)$$

$$\sum_{i=1}^{L} \alpha_{i,j} p_{i,j} \leq P_j \quad (12\text{-}4)$$

$$\sum_{j=1}^{N} \alpha_{i,j} = 1 \quad (12\text{-}5)$$

In [8], the authors have proved that if $g_{i,j}$, $r_{i,j}$, $b_{i,j}$ and $p_{i,j}$ are constant, the optimization problem of P2 is a Multiple-Choice Multidimensional Knapsack problem (MMKP), which is typical NP-hard[12][17]. And when $g_{i,j}$, $r_{i,j}$,$b_{i,j}$ and $p_{i,j}$ are all variables, P2 is a dynamic Multiple-Choice Multidimensional Knapsack problem (DMMKP). If DMMKP has solution in polynomial time, its corresponding MMKP should also have solution in polynomial time, so P2 is NP-hard. Because P2 is a simplified form of P1 in a special case, it is easy to obtain that P1 is also NP-hard

4. SLICE SELECTION ALGORITHM

4.1. Genetic Algorithm Flow Design

In Section III, we have proved that problem P1 is NP-hard, and it is difficult to obtain the optimal solution in polynomial time. Thus, we exploit genetic algorithm to solve the problem.

DOI: 10.1002/ett

In problem P1, an optimal access matrix $A = \{\alpha_{i,j}^{k}\}_{L\times N\times M}$ of the user service needs to be obtained by solving an optimization model. Correspondingly, in the genetic algorithm, we model the individual in the population as the user service access matrix *A* in the network, where *A* is a three-dimensional matrix. For a given value of *i*, only one element in the *j-k* plane has a value of 1, and the rest are all zero. A large number of "0" elements occupy a lot of storage space and increase the complexity of the problem. Therefore, we first simplify the analysis by reducing the three-dimensional matrix $A = \{\alpha_{i,j}^{k}\}_{L\times N\times M}$ to a two-dimensional matrix $A_m = \{\alpha_{i,j}\}_{L\times(N+M)}$. According to the definition of $A = \{\alpha_{i,j}^{k}\}_{L\times N\times M}$, it can be known that for the *i-th* row of A_m, only one of the first N elements has a value of 1 and others have a value of 0, i. e. $\sum_{j=1}^{N}\alpha_{i,j} \le 1$; Similarly, only one of the *(N+1)-th* to *(N+M)-th* elements has a value of 1 and the others are 0, i. e.$\sum_{j=N+1}^{N+M}\alpha_{i,j} \le 1$. Thus, the individual in the genetic algorithm are two-dimensional matrices, and the matrix elements are called chromosomes or genes. The flow chart of the genetic algorithm is shown in Fig. 3, and we will next elaborate each step of the algorithm.

4.1.1. Population Initialization and Individual Fitness

First, the number of individuals in the population are set to H, which remains constant at all times during the operation of the genetic algorithm. The population size will affect the complexity and performance of the algorithm. Therefore, in practical applications, we need to select an appropriate H value, and make a good tradeoff between the computational complexity and performance of the genetic algorithm. Let λ denote the generation of the population, which increases by one after each generation undergoes an evolution iteration, and let the initial stage λ=0. Therefore, $Q_\lambda = \{A_m^{1,\lambda}, A_m^{2,\lambda}, ..., A_m^{H,\lambda}\}$ can be used to represent the $\lambda-\mathrm{th}$ generation of the population. Furthermore, according to the constraint of $\alpha_{i,j}$, the initial values are randomly selected for chromosomes in each individual in the initial population Q_0 .

In the genetic algorithm, fitness is used to measure the merits of different individuals. In P1, the optimization objective is the overall satisfaction degree of the network. Therefore, we use the overall satisfaction degree of the network, whose access method corresponds to the access matrix $A_m^{h,\lambda}$, as the fitness of the h-th individual in the $\lambda-\mathrm{th}$ generation, which is given by

$$\eta(A_m^{h,\lambda}) = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=1}^{M}(\alpha_{h,\lambda})_{i,j}^{k} f_{i,j}^{k} \quad (13)$$

where $(\alpha_{h,\lambda})_{i,j}^{k}$ is the corresponding matrix element after the two-dimensional matrix $A_{\mathrm{m}}^{h,\lambda}=\{(\alpha_{h,\lambda})_{i,j}\}_{L\times(N+M)}$ is restored to the three-dimensional access matrix $A^{h,\lambda} = \{(\alpha_{h,\lambda})_{i,j}^{k}\}_{L\times N\times M}$.

4.1.2. Selection and Reproduction

After the initialization and individual fitness calculations, the survival of the fittest in the genetic process can be simulated by selection and reproduction. To this end, we combine probability selection and elite selection. For population Q_λ, probability selection is first performed, that is, an individual is selected from Q_λ for reproduction with a certain probability which is taken as an element that makes up population Q'_λ. This process is repeated H times, and the resulting Q'_λ has equal population size with Q_λ. In each selection, the probability of selecting an element in Q_λ follows the principle of roulette. In other words, the probability of selecting individual $A_{\mathrm{m}}^{h,\lambda}$ is proportional to its fitness, i.e.

$$P(A_m^{h,\lambda}) = \frac{\eta(A_m^{h,\lambda})}{\sum_{h=1}^{H}\eta(A_m^{h,\lambda})} \quad (14)$$

After probability selection, the elite selection is applied to improving the population Q'_λ. Specifically, the largest individual fitness $\eta'_{\lambda,\max}$ in population Q'_λ is first calculated. Then all the individuals in Q_λ whose individual fitness is greater than $\eta'_{\lambda,\max}$ are found. Replacing the same number of individuals in Q'_λ randomly with these individuals results in an improved population Q'_λ.

4.1.3. Crossover

Furthermore, crossover operation is used to simulate the process of genetic recombination during natural evolution. In this paper, we perform crossover operation by exchanging rows of individual matrices. For the population Q'_λ obtained after the selection and reproduction, all individuals in Q'_λ are randomly paired. Let $A_m^{h,\lambda}$ and $A_m^{l,\lambda}$ be paired so that all corresponding rows of $A_m^{h,\lambda}$ and $A_m^{l,\lambda}$ will be exchanged with probability P_e, where P_e is a fixed constant. After all the paired individuals in Q'_λ complete the exchange of rows, a new population Q''_λ is formed.

Trans. Emerging Tel. Tech. ; **00**:1–12 © John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

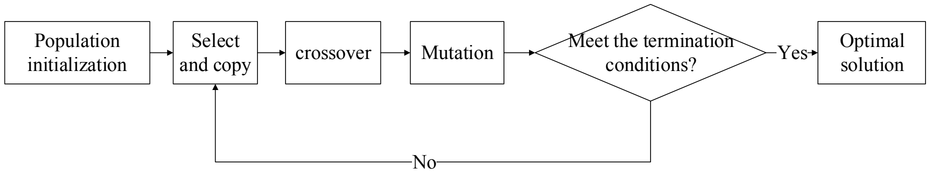

Figure 3. The flow chart of the Genetic algorithm

4.1.4. Mutation Operation

After the crossover operation, mutation operations will be performed to simulate genetic mutations in the genetic process. Similar to [13], we consider that the individual's mutation probability is determined based on the individual's fitness in the population Q''_λ. Therefore, the mutation probability of the individual $A_m^{h,\lambda}$ in Q''_λ is set to

$$P_c^{h,\lambda} = \begin{cases} \frac{\eta''_{\lambda,\max} - \eta''\left(A_m^{h,\lambda}\right)}{\eta''_{\lambda,\max} - \eta''_{\lambda,\min}} & \eta''\left(A_m^{h,\lambda}\right) < \overline{\eta''_\lambda} \\ P_c & \eta''\left(A_m^{h,\lambda}\right) \geq \overline{\eta''_\lambda} \end{cases} \tag{15}$$

where $\eta''_{\lambda,\max}$ and $\eta''_{\lambda,\min}$ are the maximum individual fitness and the minimum individual fitness in the population Q''_λ respectively. And $\overline{\eta''_\lambda}$ is the mean of the fitness of all individuals in the population Q''_λ. From (15), we can see that when the individual's fitness is greater than the average value in the population, the individual mutates with a small fixed probability P_c.

It should be noted that for $A_m^{h,\lambda}=\left\{(\alpha_{h,\lambda})_{i,j}\right\}_{L\times(N+M)}$, the chromosome of the *i-th* row needs to satisfy the constraints $\sum_{j=1}^{N}(\alpha_{h,\lambda})_{i,j} \leq 1$ and $\sum_{j=1+N}^{N+M}(\alpha_{h,\lambda})_{i,j} \leq 1$. Therefore, in the mutation operation, the mutation is performed on the row. The mutated individuals will randomly select a row of chromosomes to mutate. Specifically, the *j-th* element is first selected at random in the first N elements of the row, and set to 1, and the other N-1 elements are set to 0. Then, from the *(N+1)-th* to *(N+M)-th* elements, one element is randomly selected to be set to 1 and the remaining elements are set to 0. After performing the mutation operation on each individual in Q''_λ, the $(\lambda+1)$*th* generation population $Q_{\lambda+1}$ can be obtained.

4.1.5. Termination Rule

In genetic algorithm, the commonly used termination rule is a given maximum number of iterations T, and the algorithm terminates when the number of iterations reaches T.

4.2. Slice Selection Algorithm

According to the genetic algorithm flow described in Section 4.1, we propose a heuristic algorithm for slice selection based on genetic algorithm, as shown in Algorithm 1.

In Algorithm 1, we need to calculate the individual fitness $\eta(A_{\mathrm{m}}^{h,\lambda})$ of $A_{\mathrm{m}}^{h,\lambda}$. Moreover, according to (2) and (13), it is necessary to know the actual service rate obtained by each user corresponding to the access matrix $A_{\mathrm{m}}^{h,\lambda}$. Let the two-dimensional matrix $A_{\mathrm{m}}^{h,\lambda}=\left\{(\alpha_{h,\lambda})_{i,j}\right\}_{L\times(N+M)}$ restore to the three-dimensional access matrix $A^{h,\lambda}=\left\{(\alpha_{h,\lambda})_{i,j}^k\right\}_{L\times N\times M}$. Furthermore, we propose Algorithm 2 to calculate the service rate obtained by the user corresponding to the access matrix $A^{h,\lambda}$.

In Algorithm 2, for each user *i*, BS *k* and slice *j* can be determined according to access matrix $A^{h,\lambda}$. We consider that each slice distributes its current available power, bandwidth, and other transmission resources equally to the admitted users. Therefore, in the current scenario for user *i*, the maximum wireless rate $r_{i,j}^k$ and the service rate $c_{i,j}$ of the core network of slice can be calculated. Then we can get the maximum service rate currently available to the user $R_i = \min\{r_{i,j}^k, c_{i,j}\}$(line 3 to line 7). If $R_i \geq R_{req,i}$, let $R_{i,j}^k = R_{req,i}$. If $R_i < R_{req,i}$, it is difficult for the user to obtain a rate to meet the service requirements at this time, then user *i* is placed in set Ω_j in which users are to be admitted to accessing to slice *j* (line 13).

For each set $\Omega_j \neq \emptyset$, and for each user $h \in \Omega_j$, it is also considered to evenly distribute the current remaining available transmission resources of slice *j* to all users to be admitted. By obtaining the maximum available wireless rate $r_{h,j}^k$ and service rate $c_{i,j}$ of the core network, we can obtain the currently available maximum service rate $R_h =$

Algorithm 1 Slice Selection Algorithm

Input: i) Number of users L, number of slices N, number of BSs M.
ii) BS transmission resource limitations B_k and P_k, Slice resource limitations $B_{j,k}$, $P_{j,k}$ and C_j, $1 \leq j \leq N, 1 \leq k \leq M$.
iii) User's required rate $R_{req,i}$, $1 \leq i \leq L$, Channel gain $G_i^k, 1 \leq k \leq M$.
Output: Optimal access matrix A_{best}.
Initialization:
i) Set population size H, Crossover probability P_e, Mutation probability P_c, Maximum number of iterations T.
ii) Set λ=0, t=0 (Number of iterations), initialize population Q_0.
Calculate individual fitness $\eta(A_{\mathrm{m}}^{h,0})$ in Q_0, $1 \leq h \leq L$. Find the maximum individual fitness $\eta_{0,\max}$. Let η_{best}=$\eta_{0,\max}$, and record the corresponding optimal individual A_{best}.
repeat
  Selection and Reproduction:
  i) Select individuals from Q_λ and reproduce them with probability $P(A_m^{h,\lambda})$, and repeat H times to get the population Q'_λ.
  ii) Use the elite selection mechanism to improve Q'_λ.
  Crossover:
  Pairs all individuals in Q'_λ and exchange their rows with probability P_e to obtain population Q''_λ.
  Mutation Operation:
  Perform mutation operation to all individuals $A_m^{h,\lambda}$ in Q''_λ with probability $P_c^{h,\lambda}$ and obtain population $Q_{\lambda+1}$.
  Calculate individual fitness $\eta(A_{\mathrm{m}}^{h,\lambda+1})$ in $Q_{\lambda+1}$, $1 \leq h \leq L$. Find the maximum individual fitness $\eta_{\lambda+1,\max}$ and its corresponding optimal individual $A_{\mathrm{best}}^{\lambda+1}$. If $\eta_{\lambda+1,\max} > \eta_{best}$, let $\eta_{best} = \eta_{\lambda+1,\max}$ and $A_{best} \leftarrow A_{\mathrm{best}}^{\lambda+1}$.
  $\lambda = \lambda + 1, t = t + 1$.
until $t \geq T$
Output optimal solution A_{best}.

Algorithm 2 Computation of $R_{i,j}^k$

1: Initialize N sets $\Omega_j = \emptyset, 1 \leq j \leq N$.
2: **for** all i **do**
3: Get slice j and BS k accessed by user i according to $A^{h,\lambda}$.
4: Calculate the number of users accessing the network through the BS k and slice j $L_{j,k}$= $\sum_{i=1}^{L} \alpha_{i,j}^k$.
5: $r_{i,j}^k = \frac{B_{j,k}}{L_{j,k}} \log_2(1 + SINR_{i,j}^k)$, $SINR_{i,j}^k$=$\frac{G_i^k \cdot P_{j,k}/L_{j,k}}{I_i^k + N_0}$.
6: $c_{i,j} = \frac{C_j}{\sum_{k=1}^{K} L_{j,k}}$.
7: $R_i = \min\{r_{i,j}^k, c_{i,j}\}$.
8: **if** $R_i \geq R_{req,i}$ **then**
9: $R_{i,j}^k = R_{req,i}$.
10: C_j=$C_j - R_{req,i}$. $B_{j,k} = B_{j,k} - R_{req,i}/\log_2(1 + SINR_{i,j}^k)$.
11: $P_{j,k}$=$P_{j,k} - P_{j,k}/L_{j,k}$, $L_{j,k}$= $L_{j,k}$-1.
12: **else**
13: $\Omega_j \leftarrow \Omega_j \cup \{i\}$.
14: **end if**
15: **end for**
16: **for** each set Ω_j **do**
17: **repeat**
18: **for** $h \in \Omega_j$ **do**
19: $r_{h,j}^k = \frac{B_{j,k}}{L_{j,k}} \log_2(1 + SINR_{h,j}^k)$, $SINR_{h,j}^k$=$\frac{G_h^k \cdot P_{j,k}/L_{j,k}}{I_h^k + N_0}$.
20: $c_{h,j} = \frac{C_j}{\sum_{k=1}^{K} L_{j,k}}$.
21: $R_h = \min\{r_{h,j}^k, c_{h,j}\}$.
22: **if** $R_i \geq R_{req,i}$ **then**
23: $R_{h,j}^k = R_{req,h}$.
24: C_j=$C_j - R_{h,j}^k$. $B_{j,k} = B_{j,k} - R_{h,j}^k/\log_2(1 + SINR_{h,j}^k)$.
25: $P_{j,k}$=$P_{j,k} - P_{j,k}/L_{j,k}$, $L_{j,k}$= $L_{j,k}$-1. $\Omega_j = \Omega_j - \{h\}$.
26: **else if** $R_{req,h} > R_h \geq R_{th}$ **then**
27: $R_{h,j}^k = R_h$.
28: C_j=$C_j - R_{h,j}^k$. $B_{j,k} = B_{j,k} - R_{h,j}^k/\log_2(1 + SINR_{h,j}^k)$.
29: $P_{j,k}$=$P_{j,k} - P_{j,k}/L_{j,k}$, $L_{j,k}$= $L_{j,k}$-1. $\Omega_j = \Omega_j - \{h\}$.
30: **end if**
31: **end for**
32: **if** $\Omega_j \neq \emptyset$ **then**
33: Reject the access request of the user a with the largest $R_{req,a}$ in Ω_j.
34: $\Omega_j = \Omega_j - \{a\}$, $L_{j,k}$= $L_{j,k}$-1.
35: **end if**
36: **until** Ω_j=$\emptyset$
37: **end for**

$\min\{r_{h,j}^k, c_{h,j}\}$(line 19 to line 21). If $R_h > R_{req,h}$, $R_{h,j}^k = R_{req,h}$. If $R_{th} \leq R_h < R_{req,h}$, let $R_{h,j}^k = R_h$, where R_{th} denotes the access threshold and $R_{th} = k \cdot R_{req,h}, \quad k \in [0, 1]$. After user *h* accesses the network, *h* is removed from Ω_j and the transmission resources allocated to user *h* are removed from the available resources of slice *j* (line 22 to line 30). If $R_{th} > R_h$, the available transmission rate is less than the threshold and user *h* remains in Ω_j. After completing the calculation and judgment of each $h \in \Omega_j$, if $\Omega_j \neq \emptyset$, the access request of

DOI: 10.1002/ett

user a with the largest required rate in Ω_j is rejected (line 32 to line 34). This procedure is repeated until all $\Omega_j = \emptyset$.

5. PERFORMANCE EVALUATION

We validate the advantages of our proposed GA based slice selection algorithm by comparing it with the following access slice selection algorithms:

(1) Greedy algorithm based on SD (greedy-SD): a user selects to access the slice which can provide the maximum SD based on greedy algorithm.

(2) Greedy algorithm based on rate (greedy-rate): a user selects to access the slice which can provide the maximum rate based on greedy algorithm.

(3) RSS based on SD (RSS-SD): a user first selects the BS with highest RSS, and then selects the slice which can provide the maximum SD on this BS.

(4) RSS based on rate (RSS-rate): a user first selects the BS with highest RSS, and then selects the slice which can provide the maximum rate on this BS.

5.1. System Parameters

Table I. SIMULATION PARAMETERS

Parameter	Value
Number of BSs	4
Number of slices	5
Required rate $R_{req,i}$ (Mbps)	U[5,10]
BS transmit power $P_k^{\max}$(dBm)	47
Cell radius (m)	1060
BS wireless channel bandwidth(MHz)	20
Bandwidth of slice deployed on BS(MHz)	U[0,20]
Slice capacity(Mbps)	U[0,40]
Thermal noise (dBm/Hz)	-174
Path loss	L(d)=34+40log(d)

Table II. GENETIC ALGORITHM PARAMETERS

Parameter	Value
Maximum number of iterations T	500
Population size	50
Crossover probability	0.6
Initial mutation probability	0.01

We consider a network scenario shown in Fig. 1. Four BSs are randomly distributed in an area of $1060 \times 1060\, m^2$. Five end-to-end network slices are deployed in the network. The system parameters listed in Table 1 are similar to those used in [8]. Furthermore, the capacity C_j of the core network part of the slice j is randomly set from $[10Mbps, 40Mbps]$. BS k randomly allocates its available transmission bandwidth to all slices deployed and meets constraint $\sum_{j=1}^{5} B_{j,k} \leq 20Mbps$. It is assumed that the users are randomly distributed within the simulation range, and the required rate $R_{req,i}$ of the user is randomly generated within $[5Mbps, 10Mbps]$. Other relevant parameters of the genetic algorithm as listed in Table 2, are according to [14]. In this paper, we assume that users have the greatest SD when the service rate they receive is equal to their required rate. Therefore, let ξ=5, ρ= 1.3 in (1)-(3) [10].

5.2. Numerical Results

We first compare the performance of GA algorithm with other algorithms by using simulation experiments. Fig. 4 shows the system SD vs. the number of flows (users) in the network. As shown in the figure, we can find that GA algorithm always achieves the highest SD. When the number of users is less than 10, the network resources are sufficient, and the performance of all algorithms is close. Greedy-rate and RSS-rate, which aim at maximizing the rate, may obtain a lower satisfaction than other algorithms. This is due to the fact that the service rate is much greater than the required rate. As the number of flows increases, the proposed GA algorithm can reasonably select the slice and allocate transmission resources for users, and thus obtain higher SD than other algorithms. We can also observe that the SD of greedy-SD algorithm and RSS-SD algorithm with the optimization objective of SD is always better than the greedy-rate and RSS-rate algorithm respectively.

Furthermore, Fig. 5 shows the average Satisfaction Degree of the system. As shown in the figure, we can find that GA algorithm always achieves the best performance. As mentioned earlier, since there are sufficient network resources when the number of users is small, the greedy-rate and RSS-rate algorithm obtain service rates that are often higher than the demand rate, which results in lower Satisfaction Degree. As the number of users increases, the average rate of each user gets lower, and some users obtain a service rate close to the demand rate, so the SD increases. However, as the number of users further increases, the average rate of users decreases, resulting in a gradual decline in SD. Besides, when the network resources are sufficient, the other three algorithms with the

DOI: 10.1002/ett

optimization objective of SD obtain a service rate equals to the demand rate. As the number of users increases, the average rate of users decreases, resulting in a decrease in average Satisfaction Degree.

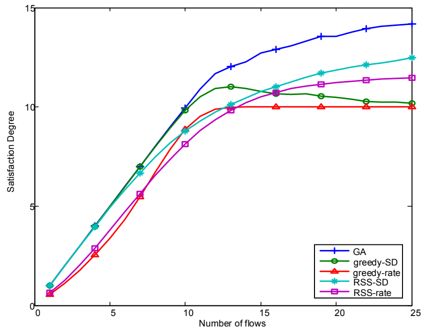

Figure 4. Satisfaction Degree of the system

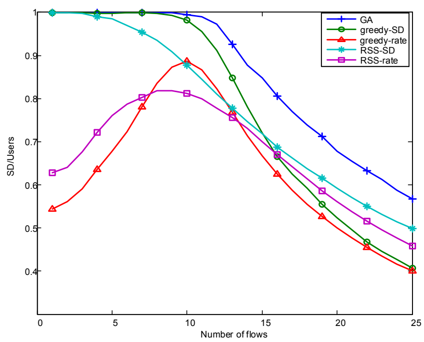

Figure 5. Average Satisfaction Degree

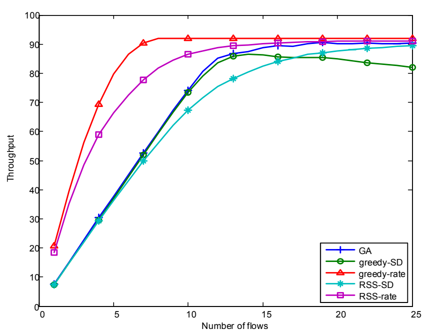

Figure 6. Throughput

Fig. 6 and Fig. 7 show the achieved throughput and average data rate versus the number of flows respectively. In GA, greedy-SD and RSS-SD algorithms

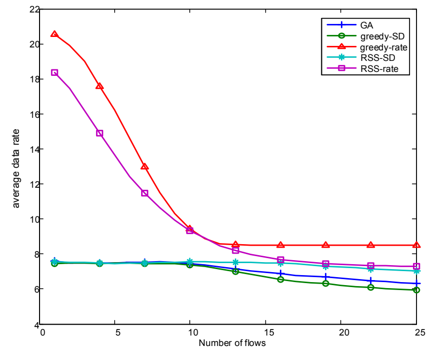

Figure 7. Average data rate

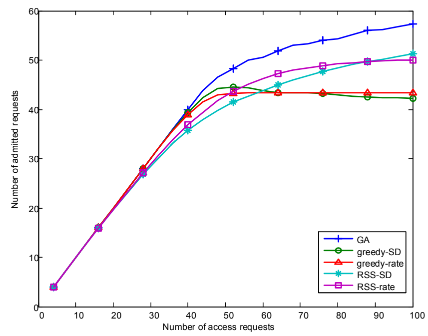

Figure 8. Number of admitted requests

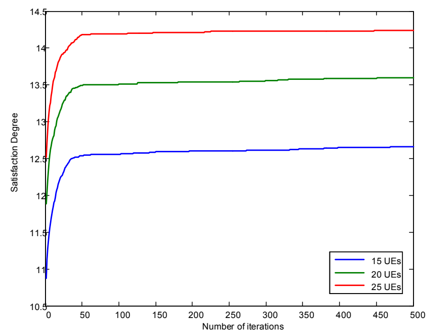

Figure 9. Convergence speed

whose optimization objective is maximizing SD, in order to obtain higher satisfaction, the service rate assigned to the users are close or equal to their demand rate when the number of flows is small and the bandwidth resources are relatively abundant. However, Greedy-rate and RSS-rate algorithms are designed to optimize the rate, and the service rate of users exceeds the demand rate. Therefore, when the number of flows is small, the system throughput

Trans. Emerging Tel. Tech. ; **00**:1–12 © John Wiley & Sons, Ltd.
DOI: 10.1002/ett
Prepared using ettauth.cls

of the Greedy-rate and RSS-rate algorithms is higher than that of the GA, Greedy-SD, and RSS-SD algorithms. As the number of flows continues increasing, the average rate of the Greedy-rate and RSS-rate algorithms gradually decreases because the available transmission resources in the network are limited. When the number of flows is large, the network tends to be saturated and the throughput of the Greedy-rate and RSS-rate algorithms can gradually approach the results of the other three algorithms.

In slice access, when the user's actual rate is lower than the access threshold $R_{th} = k \cdot R_{req,h}$, the access request will be blocked. Due to the limited network resources, blocking occurs when a slice becomes saturation. Fig. 8 compares the number of admitted slice access requests. We set the amount of transmission resource to be 4 times of that in Table 1, and set k=0.8. From the figure, we can see that the number of admitted users in GA is always higher than the other four algorithms. When the number of access requests reaches 50, the network tends to be saturated. As the GA algorithm can reasonably select the slice and allocate transmission resources for users, its advantages become more obvious. We also notice that the performance of RSS based algorithms is worse than that based on greedy when the number of access requests is small, and higher when the number is large. This is because when the number of access requests is small, the RSS-based algorithm only considers wireless channel conditions and does not consider the capacity limitation of the core network. Although BSs with better channel conditions can provide users with a higher service rate, some users may be blocked due to the limited capacity of the core network and the fact that the same core network resources are shared by the same slice at several different BSs. When the number of access requests is large, the transmission resources of the wireless are insufficient, and the service rate obtained by the users are generally smaller than the demanded rate. Compared with the greedy-based algorithm, the RSS-based algorithm always selects BSs with the best channel condition, and obtains the same wireless transmission rate using less transmission resources, so it can support more users.

Figure 9 shows the convergence speed of the GA algorithm as the number of iterations increases. When there are 15 users in the network, only about 50 iterations are needed, and the algorithm can converge. As the number of users in the network increases, the convergence time will be slightly longer.

6. CONCLUSIONS

In this paper, we have studied the wireless access selection mechanism for end-to-end network slicing from the perspective of optimal matching of users, BSs, and slices. We aim to maximize Satisfaction Degree (SD) in the system and establish a theoretical optimization model of slice selection. Through theoretical analysis, we have proved that it is NP-hard. Furthermore, we have used genetic algorithm (GA) to solve the optimization problem, and designed the corresponding slice selection optimization algorithm. On this basis, the validity of GA algorithm is verified by simulation experiment. The numerical results show that compared with the typical access selection algorithm based on the RSS or greedy algorithm in the traditional network, the GA algorithm can enable users to obtain better access and transmission performance.

REFERENCES

1. Marsch P, Da Silva I, Bulakci O, *et al*. 5G radio access network architecture: Design guidelines and key considerations. *IEEE Communications Magazine* 2016; 54(11): 24-32.
2. Lee Y L, Loo J, Chuah T C, *et al*. Dynamic Network Slicing for Multitenant Heterogeneous Cloud Radio Access Networks. *IEEE Transactions on Wireless Communications* 2018; 17(4): 2146-2161.
3. Sama M R, Beker S, Kiess W, *et al*. Service-based slice selection function for 5G. Paper presented at: 2016 IEEE Global Communications Conference (GLOBECOM); December 2016; Washington, DC USA.
4. An X, Zhou C, Trivisonno R, *et al*. On end to end network slicing for 5G communication systems. *Transactions on Emerging Telecommunications Technologies* 2017; 28(4).
5. Choyi, Vinod Kumar, *et al*. Network slice selection, assignment and routing within 5G networks. Paper presented at: 2016 IEEE International Conference

DOI: 10.1002/ett
Prepared using ettauth.cls

on Standards for Communications and Networking (CSCN); October 2016; Berlin, Germany.

6. Hu X, Chai R, Jiang G, *et al*. A joint utility optimization based virtual AP and network slice selection scheme for SDWNs. Paper presented at: 2015 IEEE International Conference on Communications and Networking in China (ChinaCom); August 2015; Shanghai, China.
7. Foukas X, Marina M K, Kontovasilis K. Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture. *In Proceedings of ACM International Conference on Mobile Computing and Networking(MobiCom)*; October, 2017; New York, USA.
8. Galeana-Zapin H, Ferrs R. Design and evaluation of a backhaul-aware base station assignment algorithm for OFDMA-based cellular networks. *IEEE transactions on wireless communications* 2010; 9(10): 3226-3237.
9. 3GPP, TS 38.300, NR and NG-RAN Overall Description, Stage 2 (Release 15), September 2017.
10. Raschell A, Bouhafs F, Seyedebrahimi M, *et al*. Quality of service oriented access point selection framework for large Wi-Fi networks. *IEEE Transactions on Network and Service Management* 2017; 14(2): 441-455.
11. Badia L, Lindstrom M, Zander J, *et al*. Demand and pricing effects on the radio resource allocation of multimedia communication systems. Paper presented at: 2003 IEEE Global Communications Conference (GLOBECOM); December 2003; San Francisco, USA.
12. Moser M, Jokanovic D P, Shiratori N. An algorithm for the multidimensional multiple-choice knapsack problem. *IEICE transactions on fundamentals of electronics, communications and computer sciences* 1997; 80(3): 582-589.
13. Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* 1994; 24(4): 656-667.
14. Wenxun Xing, Jinxin Xie, *et al*. Modern optimization calculation method, 2005.
15. Liao W C, Hong M, Farmanbar H, *et al*. Min flow rate maximization for software defined radio access networks. *IEEE Journal on Selected Areas in Communications* 2014; 32(6): 1282-1294.
16. Farmanbar H, Zhang H. Cross-layer traffic engineering for software-defined radio access networks. Paper presented at: 2015 IEEE International Conference on Communications (ICC); June 2015; London, UK.
17. Sbihi A. A best first search exact algorithm for the multiple-choice multidimensional knapsack problem. *Journal of Combinatorial Optimization* 2007; 13(4): 337-351.
18. Bouali F, Sallent O, Prez-Romero J, *et al*. Exploiting knowledge management for supporting spectrum selection in cognitive radio networks. Paper presented at: 2012 IEEE International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications(CROWNCOM); June 2012; Stockholm, Sweden.
19. Zhang H, Liu N, Chu X, *et al*. Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. *IEEE Communications Magazine* 2017; 55(8): 138-145.
20. Zhou X, Li R, Chen T, *et al*. Network slicing as a service: enabling enterprises' own software-defined cellular networks. *IEEE Communications Magazine* 2016; 54(7): 146-153.
21. 3GPP, TR 23.799 V14.0.0 (2016-12), Study on Architecture for Next Generation System.