University of Missouri, St. Louis

IRL @ UMSL

10-1-2019

Towards efficiently provisioning 5G core network slice based on resource and topology attributes

Xin Li
Wuhan University

Chengcheng Guo
Wuhan University

Jun Xu
Wuhan University

Lav Gupta
University of Missouri-St. Louis, Department of Mathematics and Computer Science,
lavgupta@missouri.edu

Raj Jain
Washington University in St. Louis

Follow this and additional works at: https://irl.umsl.edu/cmpsci-faculty

October 16, 2019

Towards Efficiently Provisioning 5G Core Network Slice Based on Resource and Topology Attributes

Xin Li
Chengcheng Guo
Jun Xu
Lav Gupta
Raj Jain

Towards Efficiently Provisioning 5G Core Network Slice Based on Resource and Topology Attributes

Xin Li [1], Chengcheng Guo [1,*], Jun Xu [1], Lav Gupta [2] and Raj Jain [3]

[1] School of Electronic Information, Wuhan University, Wuhan 430072, China; xinli1105@whu.edu.cn (X.L.); eisxujun@whu.edu.cn (J.X.)

[2] Department of Mathematics and Computer Science, University of Missouri-St. Louis, St. Louis, MO 63130, USA; lgyn6@umsl.edu

[3] Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA; jain@wustl.edu

* Correspondence: netccg@whu.edu.cn

Abstract: Efficient provisioning of 5G network slices is a major challenge for 5G network slicing technology. Previous slice provisioning methods have only considered network resource attributes and ignored network topology attributes. These methods may result in a decrease in the slice acceptance ratio and the slice provisioning revenue. To address these issues, we propose a two-stage heuristic slice provisioning algorithm, called RT-CSP, for the 5G core network by jointly considering network resource attributes and topology attributes in this paper. The first stage of our method is called the slice node provisioning stage, in which we propose an approach to scoring and ranking nodes using network resource attributes (i.e., CPU capacity and bandwidth) and topology attributes (i.e., degree centrality and closeness centrality). Slice nodes are then provisioned according to the node ranking results. In the second stage, called the slice link provisioning stage, the *k*-shortest path algorithm is implemented to provision slice links. To further improve the performance of RT-CSP, we propose RT-CSP+, which uses our designed strategy, called *minMaxBWUtilHops*, to select the best physical path to host the slice link. The strategy minimizes the product of the maximum link bandwidth utilization of the candidate physical path and the number of hops in it to avoid creating bottlenecks in the physical path and reduce the bandwidth cost. Using extensive simulations, we compared our results with those of the state-of-the-art algorithms. The experimental results show that our algorithms increase slice acceptance ratio and improve the provisioning revenue-to-cost ratio.

1. Introduction

Information and communication technologies (ICTs) are infiltrating many fields, including governance, economics, defense, media, social media, health care, industry, education, etc. [1–4]. These fields are undergoing continuous digitalization and pervasive interconnection, making communication networks an indispensable infrastructure. The coming 5G networks will promote the further upgrade of human interaction. More importantly, 5G will support a variety of vertical services, such as self-driving cars, augmented reality, live video, telemedicine, and financial transactions [5]. While 5G will improve productivity and optimize business processes, it will inevitably bring new legal and ethical issues that cannot be ignored [6,7].

The 5th generation (5G) mobile networks are expected to handle the tremendous growth of data from diverse and heterogeneous services. Softwarization, virtualization, and cloud-based

5G architecture design [8,9] are considered to be promising technologies to address the challenges introduced by the diversified service demands. Network slicing is one of the key concepts that can be realized by these techniques to support the specific needs of vertical industries. End-to-end network slicing enables multiple network services to share a single physical network infrastructure (also called the substrate network) including radio access networks (RAN) and core networks [10,11]. The big idea behind network slicing is to allow the shared 5G physical network infrastructure to be *sliced* into multiple logical networks, each of which is a collection of virtual computing and networking resources capable of supporting a specific type of service. It is, therefore, believed that network slicing will be an indispensable enabler of 5G network architecture to meet the diverse requirements of vertical applications.

　　We can broadly divide network slicing into two categories: radio access network slicing and core network slicing. In this paper, we focus on 5G core network slicing. A three-layer 5G core network slicing system model proposed by us has been elaborated in [12] and illustrated in Figure 1. There are three administrative roles in this model: 5G core infrastructure provider, 5G core slice provider, and slice tenants. The Infrastructure Provider (InP) owns the 5G core infrastructure and can lease physical resources such as computing and networking resources to the slice providers. A Slice Provider (SP) can be regarded as a virtual telecommunications service provider (TSP). The SP controls the virtualization of the resources to form network slices and provides services for users. Slice tenant is the consumer of an application specific network slice. It informs the slice provider of the characteristics of the service it needs. The slice provider requests physical resources from the infrastructure provider to create a network slice to provide the service according to the tenant's demands. The slice provisioning system interacts with the three roles to orchestrate and manage physical resources.
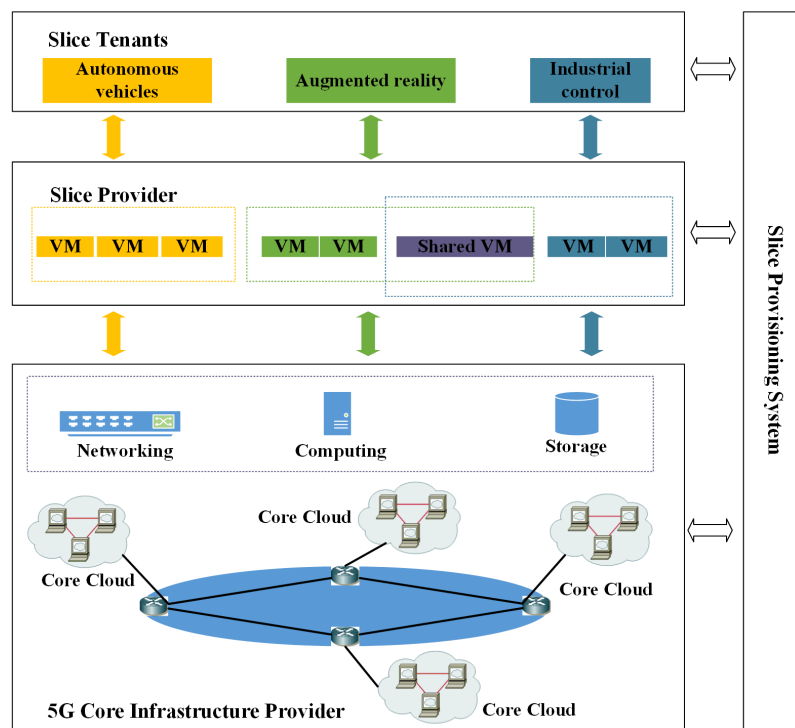
Figure 1. Illustration of three-layer 5G core network slicing system model.

　　Although network slicing has attracted increasing attention from both academia and industry [13], slice provisioning is a key issue to be addressed [14]. Slice provisioning is an approach to creating separate virtual networks based on service requirements using common physical computing and networking resources. Two sub-tasks in slice provisioning are slice node provisioning and slice link provisioning. From the perspective of InP, since the computing and networking capacities of the

physical network are limited, increasing physical resource utilization to provision more slices is crucial to raising its revenue. Therefore, in this article, we study how to efficiently provision 5G core network slices to optimize resource utilization of the 5G physical network infrastructure, thus, increasing the revenue of InP.

The slice provisioning problem in 5G network slicing is essentially the same as the traditional virtual network embedding (VNE) problem [15] in network virtualization (NV) (we rename the virtual network embedding (VNE) problem to slice provisioning problem in network slicing). Most previous VNE methods have only considered the resource attributes of the network and ignored its topology attributes to allocate physical resources to virtual network requests. Notwithstanding that several approaches consider the resource and topology attributes, the local and global resource attributes as well as the local and global topology attributes are not reasonably defined, which causes these methods to be not effective.

Based on the above considerations, we have designed a heuristic 5G core network slice provisioning strategy based on the local and global network resource attributes and topology attributes including the product of the CPU of the node and all its adjacent links, i.e., local resource attribute, the minimum bandwidth of the links in the shortest path of the node to all other nodes and the minimum CPU of the nodes along the shortest path, i.e., global resource attribute, node degree centrality, and node closeness centrality. When a 5G core slice request arrives at the slice provisioning system, the system uses resource attributes and topology attributes to perform comprehensive node evaluation and ranking, and then slice nodes are provisioned according to the ranking results. Next, slice links are provisioned using the *k*-shortest path algorithm. Our contributions are summarized as follows:

- We propose a network node scoring and ranking method by jointly considering local and global network resource and topology attributes. Specifically, we introduce a *cooperative provisioning coefficient* for the physical node scoring to enhance the efficiency of provisioning slice links.
- We design a two-stage 5G core slice provisioning algorithm, called RT-CSP, which includes a heuristic slice node provisioning algorithm and a *k*-shortest path based slice link provisioning algorithm. In the first stage, slice nodes are provisioned in a heuristic manner in accordance with the network node ranking results. In the second stage, the *k*-shortest path algorithm is used to provision slice links.
- To further improve the performance of RT-CSP, we propose RT-CSP+ slice provisioning algorithm based on our designed *minMaxBWUtilHops* strategy in the slice link provisioning stage. The strategy selects the physical path which has the minimum product of the maximum link bandwidth utilization and its hop count from the candidate physical paths obtained by the *k*-shortest path algorithm to host the slice link.
- We verify the performance of our proposed algorithm through extensive simulations and prove that our algorithm can increase the slice request acceptance ratio and, hence, the revenue of physical network provider.

The remainder of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we describe the 5G core slice provisioning problem and present the system model. The heuristic 5G core slice provisioning algorithms based on network resource attributes and topology attributes are presented in Section 4. In Section 5, we present simulation experiments and the experimental results. Finally, the conclusions and future work are laid out in Section 6.

2. Related Work

In this section, we first summarize several classic methods to solve the VNE problem with special regard to those methods considering network resource attributes or topology attributes. Then, we review the latest work related to the resource allocation in network slicing.

2.1. VNE Methods

Since VNE problem has been proven to be NP-hard by being reduced to multiway separator problem [16], its solutions can be grouped into three categories: the exact solutions, meta-heuristic solutions and heuristic solutions. Houidi et al. [17] proposed and evaluated an exact algorithm for the VNE problem using integer linear programming (ILP). They used the branch and bound method to embed virtual network requests to multiple physical networks. The results showed that the proposed exact algorithm was effective for solving small-scale problem instances. The VNE problem considered in [18] was formalized as an ILP model. Due to the complexity of ILP, the authors proposed a discrete particle swarm optimization (DPSO) method to solve the VNE problem. Simulation results showed that the DPSO method could get a better convergence performance than existing PSO methods.

Some heuristic algorithms used network resource attributes or topology attributes to rank nodes in the node mapping stage in VNE. Yu et al. [19] employed the product of the node CPU capacity and its adjacent link bandwidth for node ranking. This method was used in many other research works. However, it only considered the local resource attributes of the node. Cheng et al. [20] introduced topology attributes for the first time to embed virtual networks. Referring to the Google PageRank algorithm, they used Markov Random Walk method to rank nodes and performed node mapping in a greedy manner according to the ranking results. Wang et al. [21] introduced the network centrality in complex network theory into the VNE problem and ranked the nodes by calculating their closeness centrality. However, the closeness centrality is only one of the topology properties of the network, which measures the distance between a node and other nodes. A network topology attribute and network resource-considered algorithm was proposed to embed virtual networks in [22]. However, they only defined the local resource attributes. None of these algorithms considered the global network resource attributes as well as reasonably combined the resource attributes and topology attributes to comprehensively evaluate the importance of nodes.

2.2. Resource Allocation in Network Slicing

The authors of [23] specifically studied the problem of provisioning slice links with splittable flows. Since this problem is NP-hard, using the idea of the multipartite graph, they proposed a polynomial heuristic algorithm based on linear relaxation and randomized rounding. It was verified by simulation that the algorithm could achieve good performance. However, this study did not solve the slice node provisioning. The work in [24] used a mixed integer linear programming (MILP) model for the dynamic slicing problem and proposed several heuristic algorithms for it considering temporal variations of the virtual resource requirements. The simulation results showed that the dynamic slice provisioning could increase the slice acceptance ratio and, thus, enabled the physical network provider to increase their revenue. However, dynamic slicing comes at a cost of service quality degradation. Danish Sattar and Ashraf Matrawy [25] proposed an optimal slice allocation strategy for the 5G core network concerning the intra-slice isolation and delay requirement of slices. They formulated the problem as a MILP model and solved it with CPLEX. Their results showed that the resource utilization of the physical network would improve if the slice isolation was not considered, and stricter delay requirement also affected the slice acceptance ratio as well as resource utilization. In the very latest study [26], the authors presented a latency-optimal resource allocation method for 5G transport network slices to support URLLC services. They introduced the network resource attributes and topology attributes to resource allocation in network slicing, but they did not delve into the impact of network resource attributes and topology attributes on slice provisioning.

3. Problem Description and System Model

In this section, we first describe the 5G core slice provisioning problem, and then present the system model. A summary of used notations is presented in Table 1.

Table 1. System model notations.

Notation	Description
G^I	5G core infrastructure topological graph.
V^I	Set of physical nodes.
E^I	Set of physical links.
$c_0(v^I)$	Initial total CPU capacity of physical node v^I.
$c_a(v^I)$	Available CPU capacity of physical node v^I.
$c_u(v^I)$	Total CPU capacity of physical node v^I allocated to slice nodes.
$loc(v^I)$	Location of physical node v^I.
$\phi(v_i^I, v_j^I)$	Euclidean distance between physical nodes v_i^I and v_j^I.
$b_0(e^I)$	Initial total bandwidth of physical link e^I.
$b_a(e^I)$	Available bandwidth of physical link e^I.
$b_u(e^I)$	Total bandwidth of physical link e^I allocated to slice links.
$P^I(v_i^I, v_j^I)$	Set of loop-free physical paths between v_i^I and v_j^I.
$L(p^I(v_i^I, v_j^I))$	Set of links in $p^I(v_i^I, v_j^I)$.
G^S	5G core network slice request topological graph.
V^S	Set of slice nodes.
E^S	Set of slice links.
$c(v^S)$	CPU capability required by slice node v^S.
$loc(v^S)$	Expected deployed location of slice node v^S.
$r(v^S)$	Maximum deployed deviation allowed by slice node v^S.
$b(e^S)$	Bandwidth required by slice link e^S.

3.1. 5G Core Slice Provisioning Problem Description

The InP supplies 5G core physical infrastructure. It consists of physical nodes deployed in different locations and physical links connecting physical nodes. Physical nodes have computing, storage, and network forwarding capabilities. Virtual machine (VM) or container [27] technologies can be used to enable a physical node to host logically isolated virtual routers or VNFs (e.g., firewall, proxy, etc.). High-speed fiber optic cables are deployed in 5G core networks as physical links which have attributes such as bandwidth. Slice links are hosted on physical links or paths.

Slice tenants request 5G core network slices from the SP. A core network slice instance consists of virtual network functions and virtual links. In this study, we assume that slice nodes offer the same type of virtual network function, which is virtual computing function. Slice nodes and links request computing resources, storage resources, bandwidth resources, etc. from the slice provisioning system. Without loss of generality, we only consider computing resources and bandwidth resources here. In addition, a slice node may have a location constraint instead of being arbitrarily deployed.

The slice provisioning process includes mapping slice nodes to physical network nodes that satisfy the resource and deployment location requirements and mapping slice links onto physical paths that meet the bandwidth requests. The slice request is only accepted if the requirements of all nodes and links of the slice are satisfied; otherwise, it is rejected. We make the following assumptions for the slice provisioning problem:

- The topology of the slice remains unchanged during the life cycle of the slice, which means slice reconfiguration is not considered here.
- Slice nodes from the same 5G core network slice request can only be mapped to different physical nodes, that is, co-hosting is not allowed [28].
- Slice links cannot be split. They can only be hosted by one physical path [28].

3.2. System Model

3.2.1. 5G Core Infrastructure

The 5G core infrastructure topology is represented by a weighted undirected graph $G^I = (V^I, E^I)$, where V^I is the set of physical nodes and E^I is the set of physical links. For each physical node $v^I \in V^I$, its initial total and available computing capacities are represented as $c_0(v^I)$ and $c_a(v^I)$, respectively. Its location is denoted by $loc(v^I)$, which is represented as Cartesian coordinates, i.e., $loc(v^I) = (x(v^I), y(v^I))$. The distance of the physical link with physical nodes v_i^I and v_j^I as endpoints is the Euclidean distance between them, expressed as $\phi(v_i^I, v_j^I)$. For each physical link $e^I \in E^I$, its initial total and available bandwidth are represented as $b_0(e^I)$ and $b_a(e^I)$, respectively. The set of all loop-free paths in the infrastructure is denoted as P^I. $P^I(v_i^I, v_j^I)$ represents the set of loop-free physical paths between v_i^I and v_j^I. For each path $p^I(v_i^I, v_j^I) \in P^I(v_i^I, v_j^I)$, $L(p^I(v_i^I, v_j^I))$ is the set of links in $p^I(v_i^I, v_j^I)$. Then, the bandwidth of $p^I(v_i^I, v_j^I)$ is defined as $b(p^I(v_i^I, v_j^I)) = \min\limits_{e^I \in L(p^I(v_i^I, v_j^I))} b(e^I)$.

3.2.2. 5G Core Slice Request

5G core slice requests arrive dynamically at the resource provisioning system. The ith slice request is represented by a triplet $SR_i = (G_i^S, t_i^a, t_i^l)$, where G_i^S, t_i^a, and t_i^l represent the topology of the ith slice, its arrival time and its lifetime, respectively. The slice topology is represented by a weighted undirected graph $G^S = (V^S, E^S)$. Here, V^S is the set of slice nodes and E^S is the set of slice links. For each slice node $v^S \in V^S$, the CPU capability required by it is $c(v^S)$. Its expected deployed location is $loc(v^S) = (x(v^S), y(v^S))$ and the maximum deployed deviation allowed is $r(v^S)$, that is, the slice node can be deployed at the location within a circle whose center is the expected location $loc(v^S)$ and the radius is $r(v^S)$. Each slice link $e^S \in E^S$ is characterized by the amount of required bandwidth $b(e^S)$.

3.2.3. Slice Provisioning Process

The resource provisioning for 5G core slice task contains two sub-tasks: mapping slice nodes to physical nodes and mapping slice links to physical paths. Slice nodes from one slice request cannot be mapped to the same physical node. A slice link is mapped to a physical path with endpoints that host two slice nodes connected by the slice link. Once the resource provisioning for a slice request is successful, the allocated resource will be dedicated for the slice during its lifetime. When the lifetime ends, the allocated resource is released.

The slice node mapping function is defined as follows:

$$\boldsymbol{M}(V): \ V^S \to V', \qquad V' \subseteq V^I.$$

Slice nodes are mapped to V' that is a subset of the physical node set V^I. Since $\boldsymbol{M}(V)$ is an injective function, $\forall v_k^S, v_l^S \in V^S$, we have:

$$\boldsymbol{M}(v_k^S) = \boldsymbol{M}(v_l^S), \qquad \text{if and only if } v_k^S = v_l^S.$$

The slice link mapping function is defined as follows:

$$\boldsymbol{M}(E): \ E^S \to P', \qquad P' \subseteq P^I.$$

Slice links are mapped to P' that is a subset of all loop-free paths set P^I in the infrastructure. Thus, we define 5G core network slice mapping function as:

$$\boldsymbol{M}(S): \ (V^S, E^S) \to (V', P').$$

The calculations of resource updating during mapping process are as follows. We use $c_u(v^I, t)$ to indicate the total computing resources that the slice provisioning system assigns to all the mapped slice nodes from different slice requests at time t. It is defined below:

$$c_u(v^I, t) = \sum_{v^S} c(v^S).$$

Then, the available computing resources the physical node v^I has at time t can be calculated as:

$$c_a(v^I, t) = c_0(v^I) - c_u(v^I, t).$$

Similarly, we use $b_u(e^I, t)$ to represent the bandwidth resources the slice provisioning system has allocated to all the slice links at time t.

$$b_u(e^I, t) = \sum_{e^S} c(e^S).$$

Then, the available bandwidth the physical link e^I has at time t can be calculated as:

$$b_a(e^I, t) = b_0(e^I) - b_u(e^I, t).$$

To ensure a successful 5G core slice provisioning, all node and link constraints need to be met. The corresponding constraints are defined as follows.

Slice node mapping: Each slice node should be mapped to one physical node.

$$\sum_{v_i^I} x_i^k = 1, \qquad \forall v_k^S \in V^S, \tag{1}$$

where x_i^k indicates whether the slice node v_k^S is mapped to the physical node v_i^I or not. If v_k^S is mapped to v_i^I, x_i^k is 1. Otherwise, it is 0.

One-to-one node mapping: Each physical node can only host one slice node from the same slice request.

$$\sum_{v_k^S} x_i^k \leq 1, \qquad \forall v_i^I \in V^I. \tag{2}$$

CPU capacity: The allocated CPU capacity for slice nodes at a physical node cannot exceed the available CPU capacity of that physical node.

$$\sum_{v_k^S} x_i^k \cdot c(v_k^S) \leq c_a(v_i^I), \qquad \forall v_i^I \in V^I. \tag{3}$$

Location constraint: The distance between the mapped location of a slice node and its expected deployment location cannot exceed the maximum allowed deviation.

$$x_i^k \cdot dis(v_k^S, v_i^I) \leq r(v^S), \tag{4}$$

where $dis(v_k^S, v_i^I)$ is calculated as:

$$dis(v_k^S, v_i^I) = \sqrt{(x(v_k^S) - x(v_i^I))^2 + (y(v_k^S) - y(v_i^I))^2}. \tag{5}$$

Bandwidth: The sum of bandwidth allocated to all the slice links that are mapped to one physical link cannot exceed its available bandwidth.

$$\sum_{e_{kl}^S} y_{ij}^{kl} \cdot b(e_{kl}^S) \leq b_a(e_{ij}^I), \qquad \forall e_{ij}^I \in E^I. \tag{6}$$

If the physical link e_{ij}^I hosts the slice link e_{kl}^S, y_{ij}^{kl} is 1. Otherwise, it is 0.

3.2.4. Performance Metrics

The 5G core network infrastructure provider, while providing physical resources to the tenants, attempts to maximize its operating profit. As such, the main goal of the resource provisioning for slices is to maximize resource provisioning revenue by provisioning as many slice requests as possible. In this paper, we use slice acceptance ratio, long-term average provisioning revenue and provisioning revenue-to-cost ratio as metrics to evaluate the performance of the provisioning algorithms. They are defined below.

Slice acceptance ratio (λ): It is the ratio of the number of slices successfully provisioned to the total number of slice requests that arrive over a period of time. Then,

$$\lambda = \lim_{T \to +\infty} \frac{\sum_{t=0}^{T} S_m(t)}{\sum_{t=0}^{T} S(t)}, \tag{7}$$

where $S(t)$ is the total number of slice requests at time t and $S_m(t)$ is the number of slice requests provisioned successfully at time t.

Long-term average provisioning revenue (μ): Here, we assume the unit price of CPU capacity and bandwidth is 1. Then, the provisioning revenue of slice request G^S at time t is defined as:

$$REV(G^S, t) = \sum_{v^S \in V^S} c(v^S) + \sum_{e^S \in E^S} b(e^S). \tag{8}$$

The long-term average provisioning revenue is represented as:

$$\mu = \lim_{T \to +\infty} \frac{\sum_{t=0}^{T} \sum_{G^S \in S_m(t)} REV(G^S, t)}{T}. \tag{9}$$

Provisioning revenue-to-cost ratio (η): The provisioning cost of slice request G^S at time t is

$$COST(G^S, t) = \sum_{v^S \in V^S} c(v^S) + \sum_{e^S \in E^S} |L(p^I(e^S))| b(e^S), \tag{10}$$

where $p^I(e^S)$ is the physical path hosting the slice link e^S and $L(p^I(e^S))$ denotes the set of physical links in $p^I(e^S)$. Hence, we define provisioning revenue-to-cost ratio η as:

$$\eta = \frac{REV}{COST} = \lim_{T \to +\infty} \frac{\sum_{t=0}^{T} \sum_{G^S \in S_m(t)} REV(G^S, t)}{\sum_{t=0}^{T} \sum_{G^S \in S_m(t)} COST(G^S, t)}. \tag{11}$$

4. Heuristic 5G Core Network Slice Provisioning Algorithm Design

In this section, we describe our heuristic 5G Core Slice Provisioning algorithm based on network Resource and Topology attributes called RT-CSP in detail. Accordingly, we first present a method for ranking network nodes by using these attributes. Then, we elaborate our two-stage algorithm consisting of algorithms for heuristic slice node provisioning and *k*-shortest path based slice link provisioning. Finally, we analyze the time complexity of the RT-CSP algorithm and prove that it can run in polynomial time.

4.1. Node Ranking Based on Network Resource Attributes and Topology Attributes

In the slice node provisioning phase, a physical node for hosting a slice node needs to be carefully selected in order to meet its required CPU capacity and provisioned location requirement. Many studies in the VNE research area only consider the local resource attributes of a network node such as its CPU capacity and its adjacent link bandwidth. These works map virtual network nodes according to node ranking results based on the local resource. However, these studies do not consider global resource attributes. Moreover, the topology properties of nodes also affect the evaluation of the importance of nodes, such as degree centrality, betweenness centrality, and closeness centrality [29]. Only considering local resource attributes cannot accurately rank nodes. Therefore, we take into consideration both the local and the global resource attributes and topology attributes to comprehensively evaluate the importance of nodes.

4.1.1. Local Resource Attributes

The local resource metric of a node is obtained by multiplying the CPU capacity of the node by the sum of bandwidths of all its adjacent links.

$$LR(v_i) = c(v_i) \sum_{e \in E(v_i)} b(e),\tag{12}$$

where $E(v_i)$ is the set of all the adjacent links of v_i. The reason we define this metric is that the larger $LR(v_i)$ is, the more slice nodes can be hosted by the physical node.

4.1.2. Global Resource Attributes

Considering only the local resources of a node can cause load imbalance and resource fragmentation in the physical network. To address this, we take the minimum bandwidth of the links in the shortest path of the node to all other nodes and the minimum computing capacity of the nodes along the shortest path as the global resource metric. The following formula is its normalized definition.

$$GR(v_i) = \frac{\sum\limits_{i \neq j} [b(p(v_i, v_j)) + c(p(v_i, v_j))]}{|V| - 1},\tag{13}$$

where $b(p(v_i, v_j))$ is the minimum bandwidth of the links and $c(p(v_i, v_j))$ is the minimum CPU of the nodes in the shortest path between v_i and v_j.

4.1.3. Degree Centrality

In an undirected graph, the degree centrality of a node indicates the ratio of the number of its adjacent links to the total number of links in the graph, i.e., normalized degree centrality.

$$DC(v_i) = \frac{\sum\limits_{v_j} a_{ij}}{|V| - 1},\tag{14}$$

where a_{ij} is 1 if node v_i and node v_j are connected by a link; otherwise, it is 0. The degree centrality measures the local topological importance of the node in the network. The greater it is, the more connected the node is and the more likely it is to be selected.

4.1.4. Closeness Centrality

Closeness centrality is a method of measuring the importance of a node from a global topological perspective.

The closeness centrality of a node is obtained by first calculating the sum of the shortest paths from the node to all other nodes in the graph and then taking the reciprocal of the sum. The normalized closeness centrality is:

$$CC(v_i) = \frac{|V| - 1}{\sum\limits_{i \neq j} d(v_i, v_j)}, \tag{15}$$

where $d(v_i, v_j)$ is the length of the shortest path between node v_i and node v_j.

Thus, the nodes that are near the geometric center of the graph have higher closeness centrality.

4.1.5. Node Ranking Strategy

Our node ranking strategy combines all of the above four attributes. We rank the nodes as follows:

$$S(v_i) = \alpha LR(v_i) \cdot DC(v_i) + \beta GR(v_i) \cdot CC(v_i). \tag{16}$$

In this strategy, we have integrated the local resources, global resources, local topology attributes, and global topology attributes. It can systematically evaluate nodes in the physical network and slice requests. α and β are used to weigh the relative importance of local attributes and global attributes of the network.

4.2. Heuristic Slice Provisioning

The proposed two-stage slice provisioning algorithm is described in detail below.

4.2.1. Slice Node Provisioning

When a slice request arrives at the slice provisioning system, each slice node in the slice request is scored according to Equation (17), and then the slice nodes are ranked according to the score from high to low. The higher the score of the slice node is, the more preferentially it is provisioned. Here, $\alpha = \beta = 0.5$.

$$S(v_i) = \alpha LR(v_i^S) \cdot DC(v_i^S) + \beta GR(v_i^S) \cdot CC(v_i^S). \tag{17}$$

If physical network nodes are also scored according to Equation (17), the selected physical nodes hosting slice nodes in the final provisioning result may be far apart. Long physical paths would have to be provisioned under this circumstance, resulting in low utilization of physical network resources. To overcome this issue, we use the following cooperative provisioning method. When the current slice node is to be provisioned, the candidate physical node-set that can host the slice node is attained. The sum of the hop counts of the shortest path between the physical nodes hosting all the neighbor slice nodes of current slice node and the candidate physical node is calculated as the cooperative provisioning coefficient. The candidate physical node having the smaller coefficient may be a good hosting node. This cooperative way is beneficial to obtain a shorter physical path to host the slice link in the slice link provisioning stage, thereby improving the utilization of network bandwidth. Thus, we introduce cooperative provisioning coefficient for scoring the physical nodes.

$$H(v_i^I) = \sum_{v_j^I \in M(Adj(v^S))} h(v_i^I, v_j^I). \tag{18}$$

$$S(v_i^I) = \frac{\alpha LR(v_i^I) \cdot DC(v_i^I) + \beta GR(v_i^I) \cdot CC(v_i^I)}{H(v_i^I) + \epsilon}. \tag{19}$$

$H(v_i^I)$ is the cooperative provisioning coefficient. v_i^I represents the candidate physical node that satisfies the CPU and location requirements of slice node v^S. $M(Adj(v^S))$ represents a physical node set hosting all the neighbor slice nodes of the slice node v^S. $h(v_i^I, v_j^I)$ is the hop counts of the shortest path between the physical nodes v_i^I and v_j^I. ϵ is set to be 10^{-5} to prevent divisor from being 0. Here, $\alpha = \beta = 0.5$. The slice node provisioning algorithm is described in Algorithm 1.

Algorithm 1 Slice node provisioning based on network resource and topology attributes.

Input: Infrastructure network G^I and slice request G^S
Output: Slice node provisioning solution
 1: **for** each slice node $v^S \in V^S$ **do**
 2:    $S(v^S)$ is calculated based on Equation (17).
 3: **end for**
 4: Rank all the slice nodes in descending order of $S(v^S)$ value.
 5: Put the ranking results into *sliceNodeRankList*.
 6: **for** each physical node $v^I \in V^I$ **do**
 7:    *RTScore* based on resource and topology attributes is calculated based on Equation (16).
 8: **end for**
 9: **for** each slice node $v^S \in sliceNodeRankList$ **do**
10:    Obtain the candidate physical nodes *candidate*(v^S) for v^S meeting its CPU capacity and provisioned location demands.
11:    **if** *candidate*(v^S) is not empty **then**
12:       Obtain the physical nodes $M(Adj(v^S))$ hosting the neighbor slice nodes of v^S.
13:       **for** each physical node $v^I \in candidate(v^S)$ **do**
14:          Cooperative provisioning coefficient $H(v^I)$ is calculated based on Equation (18).
15:          $S(v^I)$ is calculated based on Equation (19).
16:       **end for**
17:       Provision v^S onto the candidate physical node which has the largest S.
18:       Put the provisioning result of v^S into *sliceNodeProvisioningList*.
19:    **else**
20:       **return** *sliceNodeProvisioningFailed*
21:    **end if**
22: **end for**
23: **return** *sliceNodeProvisioningList*

4.2.2. Slice Link Provisioning

In the slice link provisioning stage, since the slice link with a larger amount of bandwidth resource demand is more difficult to be provisioned, the slice links are first ranked according to the bandwidth requirements from large to small. The *k*-shortest path algorithm [30] is then implemented to provision slice links. Thus, our RT-CSP algorithm includes the heuristic slice node provisioning algorithm and the basic *k*-shortest path-based slice link provisioning algorithm.

To further improve the performance of RT-CSP, we propose a novel strategy, called *minMaxBWUtilHops*, in the slice link provisioning stage. After the *k*-shortest path algorithm obtains *k* candidate physical paths for each slice link satisfying its bandwidth demand, *minMaxBWUtilHops* evaluates each candidate physical path as follows:

$$\Gamma_{p^I} = (1 - \frac{b_a(e^I)}{b_0(e^I)})_{max} \cdot |L(p^I)|. \tag{20}$$

Γ_{p^I} is the product of the maximum link bandwidth utilization of the candidate physical path p^I and its hop counts. The candidate physical path with the smallest Γ_{p^I} is selected to host the slice link. The reason we propose this strategy is that the physical link with large bandwidth utilization in the physical path becomes the bottleneck of the path, which causes the provisioning of the other slice links to fail easily, resulting in a decrease in the slice acceptance ratio. In addition, selecting the physical path with fewer hops can reduce the provisioning cost. The slice link provisioning algorithm based on the *minMaxBWUtilHops* is described in Algorithm 2. The slice provisioning algorithm with heuristic slice node provisioning algorithm and the *minMaxBWUtilHops* based slice link provisioning algorithm is named as RT-CSP+.

Algorithm 2 Slice link provisioning based on *minMaxBWUtilHops*.

Input: Infrastructure network G^I, slice request G^S, and slice node provisioning results
   sliceNodeProvisioningList
Output: Slice link provisioning solution
  1: Rank all the slice links in E^S based on bandwidth requirements from large to small.
  2: Put the ranking results into *sliceLinkRankList*.
  3: **for** each slice link $e^S \in sliceLinkRankList$ **do**
  4:    k shortest path algorithm is implemented to obtain the candidate substrate paths *subPathList*
      for e^S meeting its bandwidth demand.
  5:    **if** *subPathList* is not empty **then**
  6:       **for** each substrate path $subPath \in subPathList$ **do**
  7:          Calculate Γ_{p^I} based on Equation (20).
  8:       **end for**
  9:       Provision e^S onto the candidate substrate path with the minimum Γ_{p^I}.
 10:    **else**
 11:       **return** *sliceLinkProvisioningFailed*
 12:    **end if**
 13: **end for**
 14: **return** *sliceLinkProvisioningList*

4.2.3. Slice Provisioning

When the *i*th slice request $SR_i = (G_i^S, t_i^a, t_i^l)$ arrives at the slice provisioning system, the system first checks the already provisioned slices whose lifetime ends at time t_i^a and releases the physical resources they occupied. Then, slice nodes and slice links of G_i^S are, respectively, provisioned according to the above heuristic node provisioning and the link provisioning algorithms. The slice is only accepted by the slice provisioning system if the nodes and the links are both provisioned successfully. The slice provisioning algorithm RT-CSP+ is described in Algorithm 3. The only difference between RT-CSP and RT-CSP+ is that RT-CSP uses the basic k-shortest path-based slice link provisioning while RT-CSP+ uses the *minMaxBWUtilHops* based slice link provisioning.

4.2.4. Time Complexity of RT-CSP+ Algorithm

In this section, we analyze the time complexity of the RT-CSP algorithm. Its time complexity is the sum of time complexities of slice node provisioning algorithm (i.e., Algorithm 1) and the slice link provisioning algorithm (i.e., Algorithm 2). The complexity of Algorithm 1 is dominated by the calculation of closeness centrality, whose complexity is $O(|V^I||E^I| + |V^I|^2)$. Then, the

k-shortest path algorithm [30] is implemented in the slice link provisioning stage. Its complexity is $O(k|V^I|(|E^I| + |V^I|log|V^I|))$. Therefore, the time complexity of RT-CSP algorithm is $O(|V^I||E^I| + |V^I|^2) + O(k|V^I|(|E^I| + |V^I|log|V^I|))$. It can run in polynomial time.

Algorithm 3 Slice provisioning algorithm RT-CSP.

Input: Infrastructure network G^I and the *i*th slice request $SR_i = (G_i^S, t_i^a, t_i^l)$
Output: Slice provisioning result
 1: Check slice requests whose lifetime ends at t_i^a, release physical resources they occupied and update

    physical resources.
 2: Provision slice nodes of G_i^S using Algorithm 1.
 3: **if** Slice nodes of G_i^S provisioning failed **then**

 4:     **return** *sliceProvisioningFailed*

 5: **else**

 6:     Provision links of slice G_i^S using Algorithm 2.
 7:     **if** Links of slice G_i^S provisioning failed **then**

 8:         **return** *sliceProvisioningFailed*

 9:     **else**

10:         Provision slice request G_i^S, allocate physical resources and update physical resources.
11:         **return** *sliceProvisioningSucceeded*
12:     **end if**
13: **end if**

---

5. Performance Evaluation

In this section, we evaluate the performance of the proposed heuristic 5G core slice provisioning algorithms RT-CSP+ and RT-CSP. First, we describe the experimental settings for implementing our algorithms. Then, we present the results obtained from extensive evaluation experiments and analyze the results by comparing them with the state-of-the-art algorithms.

5.1. Evaluation Settings

We developed a discrete event simulator using Java to evaluate our algorithms and ran all the experiments on a Windows 10 laptop with Intel Core i7-6820HQ CPU and 24 GB RAM. The topology generation package "Brite" [31] was integrated with our simulator to generate the 5G core infrastructure topology and the 5G core slice requests based on the Waxman topology model [32].

To compare our results with those of existing research, the simulation parameters were set according to the parameter settings widely used in previous research [19,28,33]. They are described as follows and summarized in Table 2.

The physical network nodes are randomly deployed in a rectangular area of 500 by 500. The initial total available CPU capacities of the nodes are real numbers uniformly distributed between 50 and 100. Adjacent nodes are connected by a probability of 0.5 to form physical links, whose initial total available bandwidths are real numbers uniformly distributed between 50 and 100.

The 5G core slice requests arrive following a Poisson process. The number of nodes in the slice request is a uniformly distributed integer between 2 and 10. For each slice request, the slice nodes allow the provisioned position to have a deviation of less than 80. The CPU demands of the slice nodes are real numbers uniformly distributed between 1 and 20. Slice nodes are connected by a probability to form slice links. The bandwidth requirement of each slice link takes a uniformly distributed real number in the range [1, 20]. The lifetime of the slice request follows the exponential distribution with a mean of 500 time units. We have 2000 slice requests in total in the experiments.

Table 2. Notations of system model.

Parameter	Description
Number of substrate nodes	50/100/150
Probability of connecting substrate nodes	0.5
Substrate node CPU	$U[50, 100]$
Substrate link bandwidth	$U[50, 100]$
Lifetime of slice requests obeying Exponential distribution	500 time units in average
Number of slice requests	2000
Number of slice nodes in each slice	$U[2, 10]$
Probability of connecting slice nodes	0.2/0.5/0.8
Slice node CPU demand	$U[1, 20]$
Slice link bandwidth demand	$U[1, 20]$

5.2. Evaluation Results and Analysis

To evaluate the experimental results, we compared the state-of-the-art algorithms, as listed in Table 3. The RT-CSP+ and RT-CSP algorithms are our proposed algorithms. First, we evaluated the performances of these algorithms in the scenario where the slice request arrival rate is four requests per 100 time units. Next, we changed the slice link connected probability to study its effects on the performance of the algorithms. Then, to verify the scalability of our algorithms, we examined simulation scenarios with different slice arrival rates and different sizes of the substrate network. We ran each experiment for 10 times to analyze experimental results.

Table 3. Algorithms for comparison.

Notation	Description
RT-CSP+	The provisioning algorithm considering resource and topology attributes with the *minMaxBWUtilHops* based slice link provisioning
RT-CSP	The provisioning algorithm considering resource and topology attributes with the basic *k*-shortest path-based slice link provisioning
VNE-DCC	The algorithm considering local resource and topology attributes in [33]
NRM-VNE	The algorithm only considering local resource attributes in [28]
CC	The provisioning algorithm in [21] considering classic closeness centrality

5.2.1. Experiments in the Scenario where the Slice Request Arrival Rate Is Four Requests Per 100 Time Units

In this scenario, there are 100 substrate nodes in the substrate network and the slice nodes are connected by a probability of 0.5. The results of slice acceptance ratio, long-term average revenue and the revenue-to-cost ratio of the algorithms are shown in Figures 2 and 3a,b, respectively.

Figure 2 shows our algorithm RT-CSP+ has the best slice acceptance performance over the entire simulation time. The acceptance ratio of all algorithms is relatively high at the beginning of the simulation because the CPU and bandwidth of the physical network are sufficient. As the simulation progresses, the available resources of the physical network gradually reduce due to the occupation of the active slice requests in the provisioning system, resulting in a gradual decrease in the slice reception ratio. After 10,000 time units, the slice acceptance ratio tends to stabilize. The reason is that the arrival and departure of the slices reach a relatively balanced state, and thus the available resources of the physical network are relatively stable. When the simulation time reaches 40,000 time units, the slice acceptance ratio of RT-CSP+ is 91.52%, which is 15.06%, 17.93%, and 51.25% higher than those of VNE-DCC, NRM-VNE, and CC, respectively. Our algorithm can comprehensively evaluate nodes from the perspective of local and global resource and topology attributes, making node provisioning more optimized. Thus, our algorithm can increase the slice acceptance ratio. The slice acceptance

ratio of RT-CSP+ is higher than RT-CSP, which shows that our *minMaxBWUtilHops* strategy in the link provisioning stage can further enhance the performance of RT-CSP.
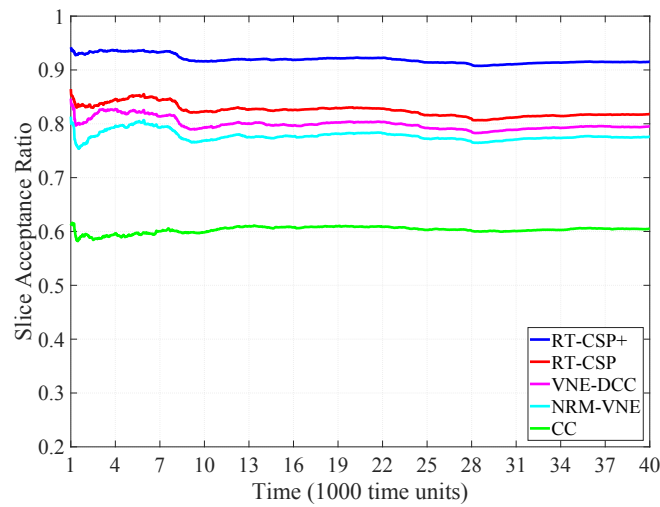

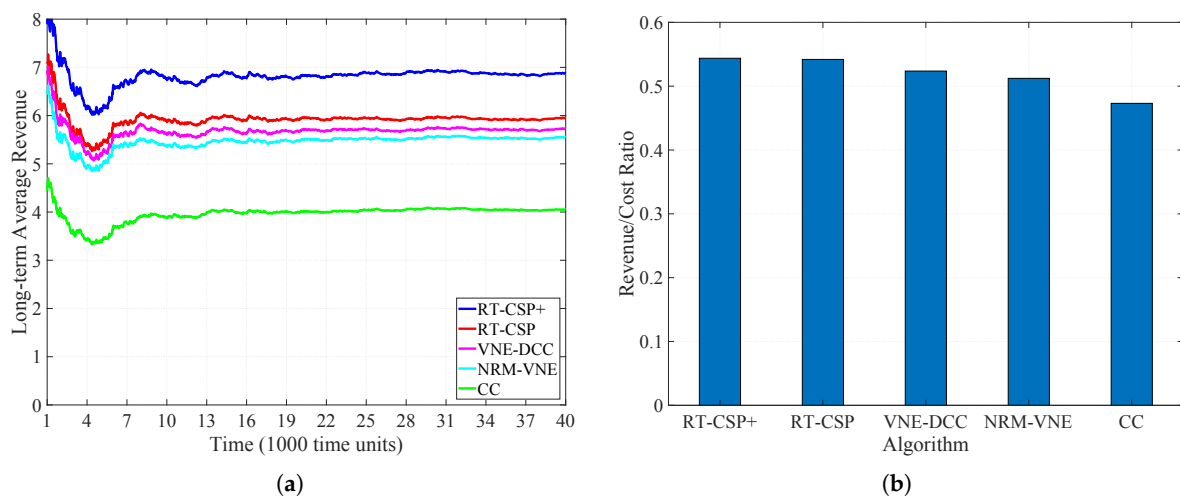**Figure 2.** Slice acceptance ratio.

Figure 3. (**a**) Long-term average slice provisioning revenue; and (**b**) slice provisioning revenue-to-cost ratio.

As shown in Figure 3a, the RT-CSP+ algorithm has the largest long-term average slice provisioning revenue. In the early stage of the simulation, the long-term average revenue decreases rapidly. The reason is that, as the slice arrives, the physical resources are consumed. The subsequent arriving slices are easy to be rejected, which decreases the provisioning revenue. When the simulation time reaches 10,000 time units, it tends to be stable because the arrival and departure of the slices reach a relatively balanced state. In the final steady state, the long-term average revenue of RT-CSP+ algorithm is 20.01%, 23.86% and 69.88% higher than those of VNE-DCC, NRM-VNE, and CC, respectively. Similar to the slice acceptance ratio and the long-term average revenue, the revenue-to-cost ratio also tends to be stable after 10,000 time units. Therefore, we show the average revenue-to-cost ratio histogram during the steady stage in Figure 3b. The RT-CSP+ and RT-CSP algorithms have better performance than others in terms of this metric. This is consistent with the long-term average revenue performance. Furthermore, since the revenue-to-cost ratio depends on the revenue and cost, the larger revenue-to-cost ratio is not only because our algorithms can achieve higher revenue, but also because it can reduce the provisioning cost.

5.2.2. Experiments in the Different Slice Link Connected Probability Scenario

We experimented on the different slice link connected probability scenario, in which the slice link connected probability is 0.2, 0.5, and 0.8, respectively, to investigate its impact on the performance of the algorithms. Figures 4 and 5 present the results of the slice acceptance ratio and slice provisioning revenue performance.
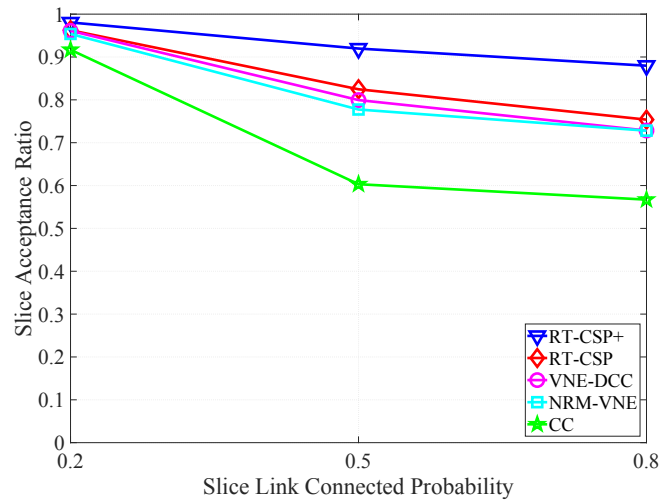

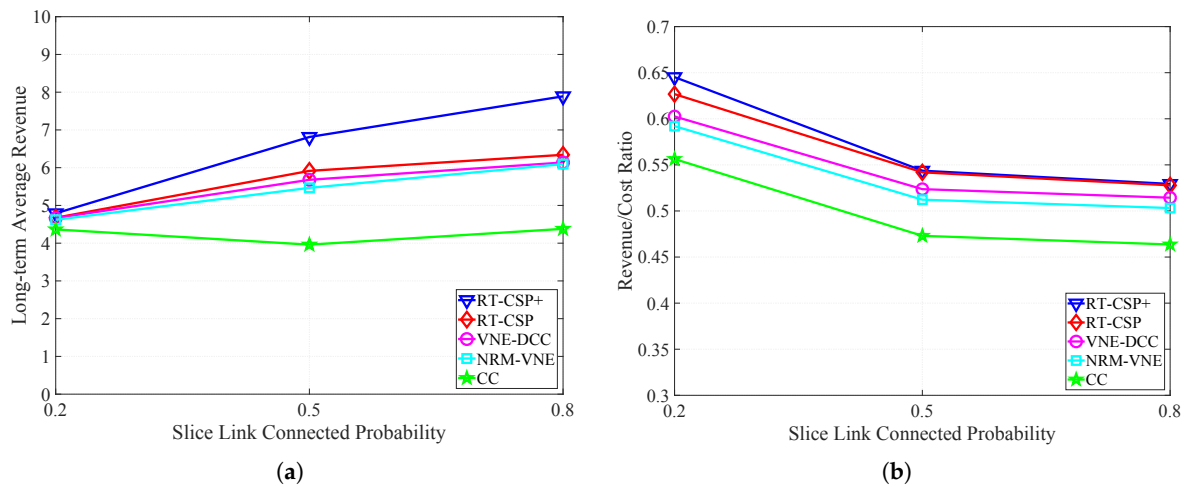**Figure 4.** Slice acceptance ratio in the different slice link connected probability scenario.

Figure 5. (**a**) Long-term average slice provisioning revenue in the different slice link connected probability scenario; and (**b**) slice provisioning revenue-to-cost ratio in the different slice link connected probability scenario.

Figure 4 shows that the slice acceptance ratio decreases as the slice link connection probability increases. This is because slice requests with more slice links demand more bandwidth resources, which makes the physical network difficult to satisfy bandwidth demands, resulting in more rejected slice requests. On the other hand, RT-CSP+ always has the highest slice acceptance ratio because of its efficiency. Figure 5a shows that, as the slice link connection probability increases, the long-term average revenue of all algorithms increases except for that of CC. For algorithms except CC, although the slice acceptance ratio is smaller at larger slice link connection probability, more slice links are provisioned in this case, which brings more provisioning revenue. For CC, when the slice link connection probability is 0.2, it can obtain much better slice acceptance ratio compared with 0.5 and 0.8, which contributes a lot to provisioning revenue. The long-term average revenue of CC has a similar trend as other algorithms when the slice link connection probability gets larger. With regard to the long-term average

provisioning revenue, RT-CSP+ still outperforms others. Figure 5b shows that the revenue-to-cost ratio decreases as the slice link connection probability increases. The reason more revenue cannot result in larger revenue-to-cost ratio is that more slice links should be provisioned when the slice link connection probability is larger, in which case slice links are easier to be provisioned to a longer physical path, resulting in more provisioning bandwidth cost.

5.2.3. Experiments in the Different Slice Request Arrival Rates Scenario

We further validated the performance of our proposed algorithm by experimenting with different slice arrival rates. There are 100 substrate nodes in the substrate network and the slice nodes are connected by a probability of 0.5 in this scenario. Figures 6 and 7 show the results of the slice acceptance ratio and slice provisioning revenue performance with mean slice arrival rates of 0.02, 0.04, 0.06, 0.08, and 0.1.
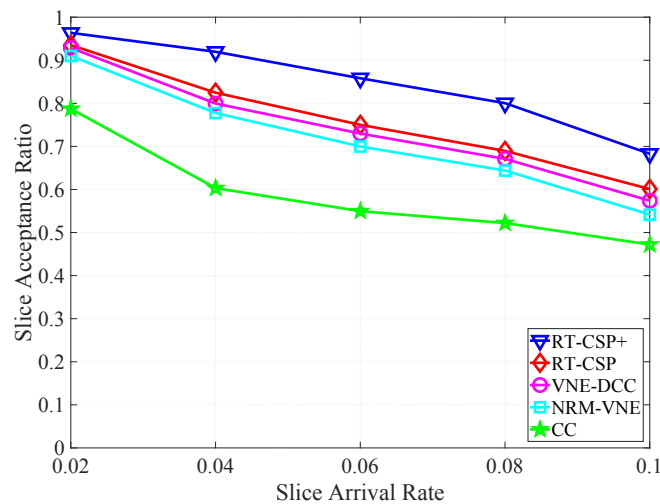

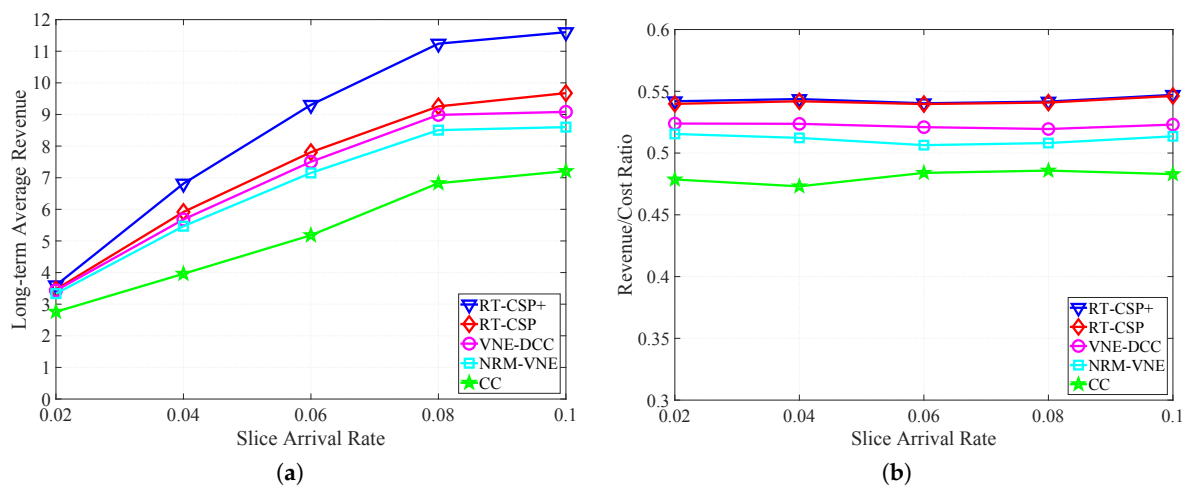**Figure 6.** Slice acceptance ratio in the different arrival rates scenario.

Figure 7. (**a**) Long-term average slice provisioning revenue in the different arrival rates scenario; and (**b**) slice provisioning revenue-to-cost ratio in the different arrival rates scenario.

As can be seen in Figure 6, RT-CSP+ algorithm always has the highest slice acceptance ratio when slices arrive at different rates. For example, when the slice request arrival rate is 0.06, the slice acceptance ratio of RT-CSP+ is 85.82%, which is 14.43%, 17.59%, 22.65%, and 56.15% higher than those of RT-CSP, VNE-DCC, NRM-VNE, and CC, respectively. This is because RT-CSP+ can comprehensively optimize node provisioning using the resource and topology attributes and the *minMaxBWUtilHops*

strategy increases the probability of successfully provisioning slice links. In addition, slice acceptance ratio of all algorithms decreases as the slice arrival rate increases. The reason is that the larger the slice arrival rate, the more slices enter the slice provisioning system per unit time. Due to the limited physical resources, the probability of slice provisioning failure increases when more slices compete for limited physical resources, resulting in low slice acceptance ratio.

Figure 7a shows that the RT-CSP+ and RT-CSP algorithm always have better long-term average slice provisioning revenue with different slice arrival rates. This is because RT-CSP+ and RT-CSP can reasonably evaluate nodes in the node provisioning stage, resulting in more slices to be received. For each algorithm, the reason the long-term average revenue grows as the arrival rate increases is that more slice requests arrive per time unit under higher arrival rate scenario. Thus, more revenue can be obtained per time unit. Figure 7b presents that the average slice provisioning revenue-to-cost ratio during the steady stage is relatively stable with different slice arrival rates because the arrival and departure of the slices can reach a relatively balanced state. Our algorithms still have higher revenue-to-cost ratio.
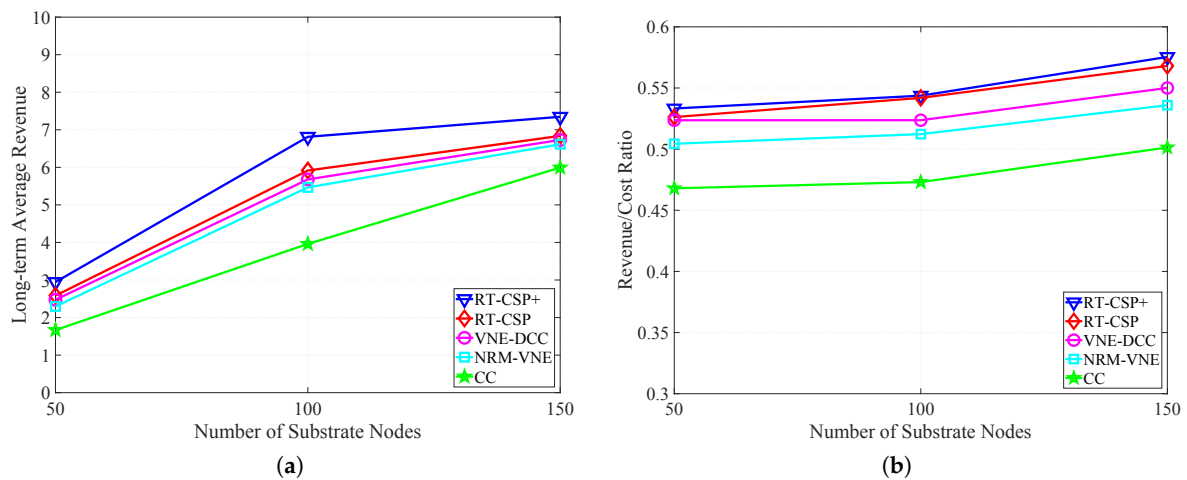
5.2.4. Experiments in The Different Sizes of Substrate Network Scenario

The slice nodes are connected by a probability of 0.5 in this scenario. Figures 8 and 9 show the results of the slice acceptance ratio and slice provisioning revenue performance when the number of substrate network nodes is 50, 100, and 150, which represent small-, medium-, and large-sized physical network, respectively.

Figure 8 shows that, when the size of the physical network gets larger, all the algorithms have higher slice acceptance ratio. This is because the physical network with larger size has sufficient resources to host slice requests, which makes it easier to accept more slice requests. In the scenario with different sizes of substrate network, RT-CSP+ always has best slice acceptance ratio. For instance, when the substrate network has 150 nodes, the slice acceptance ratio of RT-CSP+ is 98.30%, which is 6.02%, 6.58%, 7.92%, and 17.44% higher than those of RT-CSP, VNE-DCC, NRM-VNE, and CC, respectively. The reason is that RT-CSP+ can efficiently provision slice requests based on the resource and topology attributes. In accordance with better slice acceptance ratio, our algorithms can produce better revenue performance as shown in Figure 9. From another aspect, the better performance of our algorithm in this scenario verifies its the scalability.
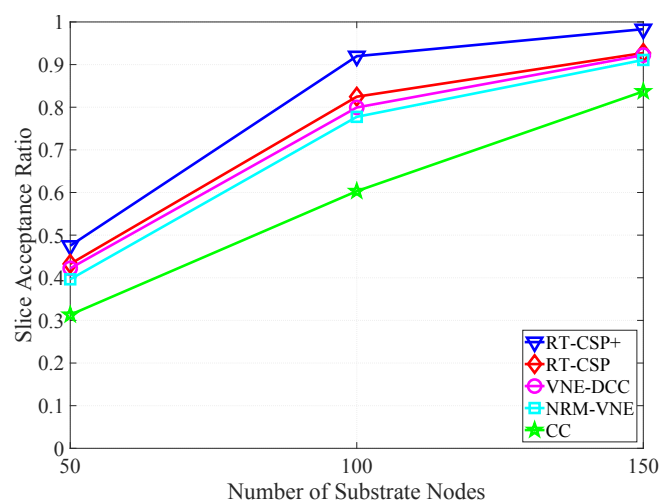
Figure 8. Slice acceptance ratio in the different sizes of substrate network scenario.

Figure 9. (**a**) Long-term average slice provisioning revenue in the different sizes of substrate network scenario; and (**b**) slice provisioning revenue-to-cost ratio in the different sizes of substrate network scenario.

6. Conclusions

5G will be a disruptive technology in many ways. It has the potential to shakeup the telecommunications industry but would require significant investments. Consumers, both businesses and individuals, expect new opportunities from massive, ultra low latency and high density Internet of Things, as a run up to ambitious use cases such as smart cities and autonomous vehicles. The potential of 5G can only be truly realized if telecommunications service providers build in economies in the new deployments. Network slicing would be a key factor in achieving increased efficiencies and revenues through service specific offerings.

We have worked on the slice-provisioning problem by taking into account both the slice node provisioning and the slice link provisioning aspects. Accordingly, we have proposed a two-stage slice-provisioning algorithm called RT-CSP. As far as provisioning of slice nodes is concerned, our method takes into account the compute capacities, link bandwidths, degree centrality, and closeness centrality for comprehensive evaluation and ranking of nodes. These amount to jointly considering the local and global network resource attributes along with the topology attributes. Along with the heuristic slice node provisioning algorithm, RT-CSP uses the *k*-shortest path based slice link provisioning algorithm. An enhancement developed by us called RT-CSP+, based on *minMaxBWUtilHops* strategy designed by us, improves the performance further by selecting the physical path that has the minimum product of the maximum link bandwidth utilization and its hop count from the candidate physical paths obtained by the *k*-shortest path algorithm.

Extensive evaluations were carried out to compare both of our algorithms with other state-of-the-art algorithms and prove that the proposed algorithm does increase the slice request acceptance ratio and consequently the revenue of the network infrastructure provider. As far as acceptance ratio is concerned, both RT-CSP and RT-CSP+ perform better than other algorithms with the latter consistently giving the best performance. As the slice request arrival rate increases, the acceptance ratio of all the algorithms goes down but RT-CSP+ retains its supremacy. In terms of provisioning revenue, RT-CSP+ excels in long-term average slice provisioning revenue and revenue-to-cost ratio. Both RT-CSP and RT-CSP exhibit better revenue performance than other algorithms as the arrival rate increases. These results verify that our algorithms can comprehensively optimize node provisioning using the resource and topology attributes.

We are enthused with the good performance of our algorithms and, in the future, we plan to propose an efficient provisioning solution for latency-sensitive slices to satisfy low-latency 5G applications.

References

1. Amendola, C.; Calabrese, M.; Caputo, F.; Fabrizio, D. Fashion companies and customer satisfaction: A relation mediated by Information and Communication Technologies. *J. Retail. Consum. Serv.* **2018**, *43*, 251–257. [CrossRef]

2. Del Giudice, M.; Caputo, F.; Evangelista, F. How are decision systems changing? The contribution of social media to the management of decisional liquefaction. *J. Decis. Syst.* **2016**, *25*, 214–226. [CrossRef]

3. Stone, D.L.; Deadrick, D.L.; Lukaszewski, K.M.; Johnson, R. The influence of technology on the future of human resource management. *Hum. Resour. Manag. Rev.* **2015**, *25*, 216–231. [CrossRef]

4. Susskind, R.E.; Susskind, D. *The Future of the Professions: How Technology Will Transform the Work of Human Experts*; Oxford University Press: New York, NY, USA, 2015.

5. Papagiannis, H. *Augmented Human: How Technology Is Shaping The New Reality*; O'Reilly Media, Inc.: Farnham, UK, 2017.

6. Selin, C. The Ethics of Invention Technology and the Human Future. *Science* **2016**, *353*, 756. [CrossRef]

7. Johnson, D.G. Technology with no human responsibility? *J. Bus. Ethics* **2015**, *127*, 707–715. [CrossRef]

8. Rost, P.; Banchs, A.; Berberana, I.; Breitbach, M.; Doll, M.; Droste, H.; Mannweiler, C.; Puente, M.A.; Samdanis, K.; Sayadi, B. Mobile network architecture evolution toward 5G. *IEEE Commun. Mag.* **2016**, *54*, 84–91. [CrossRef]

9. Jain, R.; Paul, S. Network virtualization and software defined networking for cloud computing: A survey. *IEEE Commun. Mag.* **2013**, *51*, 24–31. [CrossRef]

10. NGMN. Description of Network Slicing Concept. Available online: https://www.ngmn.org/fileadmin/user_upload/160113_Network_Slicing_v1_0.pdf (accessed on 2 May 2019).

11. 3GPP. *Study on Management and Orchestration of Network Slicing for Next Generation Network (Release 15)*; Technical Specification (TS) 28.801, 3rd Generation Partnership Project (3GPP); 3GPP: Sophia Antipolis, France, 2018 .

12. Li, X.; Samaka, M.; Chan, H.A.; Bhamare, D.; Gupta, L.; Guo, C.; Jain, R. Network slicing for 5G: Challenges and opportunities. *IEEE Internet Comput.* **2017**, *21*, 20–27. [CrossRef]

13. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [CrossRef]

14. Kim, Y.; Kim, S.; Lim, H. Reinforcement Learning Based Resource Management for Network Slicing. *Appl. Sci.* **2019**, *9*, 2361. [CrossRef]

15. Fischer, A.; Botero, J.F.; Beck, M.T.; De Meer, H.; Hesselbach, X. Virtual network embedding: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1888–1906. [CrossRef]

16. Andersen, D.G. *Theoretical Approaches to Node Assignment*; Carnegie Mellon University: Pittsburgh, PA, USA, 2002. Available online: http://repository.cmu.edu/compsci/86/ (accessed on 16 October 2019).

17. Houidi, I.; Louati, W.; Ameur, W.B.; Zeghlache, D. Virtual network provisioning across multiple substrate networks. *Comput. Netw.* **2011**, *55*, 1011–1023. [CrossRef]

18. Wang, L.; Qu, H.; Zhao, J.; Guo, Y. Virtual network embedding with discrete particle swarm optimisation. *Electron. Lett.* **2014**, *50*, 285–286. [CrossRef]

19. Yu, M.; Yi, Y.; Rexford, J.; Chiang, M. Rethinking virtual network embedding: Substrate support for path splitting and migration. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 17–29. [CrossRef]

20. Cheng, X.; Su, S.; Zhang, Z.; Wang, H.; Yang, F.; Luo, Y.; Wang, J. Virtual network embedding through topology-aware node ranking. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 38–47. [CrossRef]

21. Wang, Z.; Han, Y.; Lin, T.; Tang, H.; Ci, S. Virtual network embedding by exploiting topological information. In Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 2603–2608.

22. Cao, H.; Yang, L.; Zhu, H. Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding. *IEEE Internet Things J.* **2017**, *5*, 108–120. [CrossRef]

23. Paschos, G.S.; Abdullah, M.A.; Vassilaras, S. Network Slicing with Splittable Flows is Hard. In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1788–1793.

24. Raza, M.R.; Fiorani, M.; Rostami, A.; Öhlen, P.; Wosinska, L.; Monti, P. Dynamic slicing approach for multi-tenant 5G transport networks. *J. Opt. Commun. Netw.* **2018**, *10*, A77–A90. [CrossRef]

25. Sattar, D.; Matrawy, A. Optimal Slice Allocation in 5G Core Networks. *arXiv* **2018**, arXiv:1802.04655.

26. Li, W.; Zi, Y.; Feng, L.; Zhou, F.; Yu, P.; Qiu, X. Latency-Optimal Virtual Network Functions Resource Allocation for 5G Backhaul Transport Network Slicing. *Appl. Sci.* **2019**, *9*, 701. [CrossRef]

27. Bernstein, D. Containers and cloud: From lxc to docker to kubernetes. *IEEE Cloud Comput.* **2014**, *3*, 81–84. [CrossRef]

28. Zhang, P.; Yao, H.; Liu, Y. Virtual network embedding based on computing, network, and storage resource constraints. *IEEE Internet Things J.* **2017**, *5*, 3298–3304. [CrossRef]

29. Newman, M. *Networks*; Oxford University Press: New York, NY, USA, 2018.

30. Yen, J.Y. Finding the k shortest loopless paths in a network. *Manag. Sci.* **1971**, *17*, 712–716. [CrossRef]

31. Medina, A.; Lakhina, A.; Matta, I.; Byers, J. *BRITE: Universal Topology Generation from a User's Perspective*; Technical Report; Boston University Computer Science Department: Boston, MA, USA, 2001.

32. Waxman, B.M. Routing of multipoint connections. *IEEE J. Sel. Areas Commun.* **1988**, *6*, 1617–1622. [CrossRef]

33. Zhang, P.; Yao, H.; Liu, Y. Virtual network embedding based on the degree and clustering coefficient information. *IEEE Access* **2016**, *4*, 8572–8580. [CrossRef]