Enforcing Resource Allocation and VNF Embedding in RAN Slicing

Kashyab J. Ambarani*, Shivansh Sharma*, Shravan Komarabattini*, My T. Thai†, and Tu N. Nguyen‡

*Department of Computer Science, Purdue University Fort Wayne, IN 46805, USA
†Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32612, USA
‡Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA
Email: *{ambakj01, shars03, komas01}@pfw.edu, †mythai@cise.ufl.edu, ‡tu.nguyen@kennesaw.edu.

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685958

***Abstract*—Going beyond the *one-type-fits-all* design philosophy, the future 5G radio access network (RAN) with *network slicing methodology* is employed to support widely diverse applications over the same physical network. RAN slicing aims to logically split an infrastructure into a set of self-contained programmable *RAN slices* in which each slice built on top of the underlying physical RAN (substrate) is a separate logical mobile network and delivers a set of services with similar characteristics. Each RAN slice is constituted by various virtual network functions (VNFs) distributed geographically in numerous substrate nodes. A RAN configuration scheme for the network is imperative to embed VNFs in substrate nodes. In this paper, we propose to design new algorithms to enhance the stability of RAN slicing by addressing the resources allocation and VNF embedding problem, referred to as *RS-configuration*. Specifically, we establish the theoretical foundation for using RS-configuration to construct a VNF mapping plan for all VNFs with two efficient algorithms, including Group-based Algorithm (GBA) and Group-Connectivity-based Algorithm (GCBA). Through rigorous analysis and experimentation, we demonstrate that the proposed algorithms perform well within reasonable bounds of computational complexity.**

***Index Terms*—RAN slicing, VNF embedding, resource allocation.**

I. Introduction

With ever more devices connected to the Internet and the creation of new services, such as mobile broadband, video streaming, the massive Internet of Things, and autonomous vehicle management, demands for diverse vertical industry applications are growing rapidly to expand the wireless market. To address the various application demands, Radio Access Network (RAN) slicing has become one of the most promising architectural technologies for the forthcoming 5G era [1], [2]. RAN slicing completely overturns the traditional model of a single ownership of all network resources and brings a new vision where the physical infrastructure resources are shared across many *RAN slices*. Each slice built on top of the underlying physical RAN (substrate) is a separate logical mobile network, which delivers a set of services with similar characteristics and is isolated from others [3], [4]. Leveraged by network function virtualization (NFV), a RAN slice is constituted by various virtual network functions (VNFs) and *virtual links* that are embedded as instances on substrate nodes [5]. RAN enforcement mechanisms enable a highly efficient resource management service and maximizes the resources configured [6]. Efficient resource allocation and VNF embedding serve as one of the critical aspects in RAN slicing technologies. The resource allocation and VNF embedding problem is considered under a variety of constraints [6].

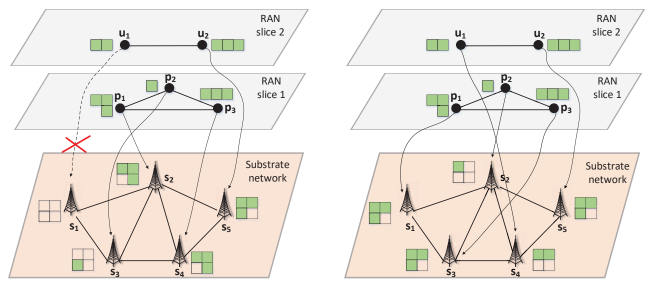
Motivation. Conventional configuration schemes[1] in RAN slicing that underpin much of exiting designs are primarily *resource allocation-based*, in which only allocating resources of substrate nodes to VNFs is considered [7], [8], [9]. However, a configuration scheme will not only hinge on substrate nodes resource allocation but also rely on the connection between the substrate nodes. Since there is a specific connection between VNFs, it thus requires a corresponding connection between substrate nodes when the VNFs are mapped onto the nodes, and, obviously the mappings are also bounded by the bandwidth requirement of the virtual links between VNFs [10] (see §II-A). Furthermore, a configuration process comes at an operating cost, delay, and high impact to the entire network. Thus, most existing RAN enforcement algorithms for RAN slicing do not explicitly consider the requirements of RAN slices (e.g., bandwidth and connectivity requirements) while mapping VNFs, nor do they attempt to consider the interdependency property to mitigate the impact of the configuration process on the entire network.

Contributions. Going beyond resource allocation-based configuration schemes, in this paper we advocate a novel configuration scheme−dubbed *RAN slicing-configuration (RS-configuration)*. Mathematically, a RS-configuration is an ordered set of mappings of VNFs onto substrate nodes. Instead of considering only available resources, we consider the interdependence between all *possible* mappings and rearrange resource allocation to attain a higher VNF embedding performance what we refer to as a (mapping plan) *RS-configuration* for RAN slicing. RS-configuration also aims to mitigate the impact of the configuration process on the entire network and enable us to analyze and reason *VNF embedding* and *RAN configuration* trade-offs.

Tu N. Nguyen was with the Department of Computer Science at Purdue University Fort Wayne at the time of submission.

[1]Some other works refer to them as RAN enforcement.

978-1-7281-8104-2/21/$31.00 ©2021 IEEE



(a) (b)

Fig. 1: Resources allocation and embedding VNFs in a RAN slicing with a) a random mapping and b) RS-configuration algorithms (best viewed in color).

The proposed research will focus on two fundamental tasks: i) establishing the theoretical foundation for using RS-configuration to construct a VNF mapping plan for RAN slice embedding optimization; ii) developing a scheme and algorithms needed to map/embed VNFs efficiently. In the following sections, the PI describes the specific context and contributions of this project and the proposed plan for completing this work.

Organization: The paper is organized as follows. In §II, we present the network model of the network slicing. In this section, we also discuss the problem definition. We present related formulation, metric, and resource allocation algorithms in §III. In §IV, we evaluate the performance of the proposed algorithms. Finally, we make key concluding remarks in §V.

II. Network Model and Research Problem

In the following sections, the basic mathematical notations are initially introduced and simple examples are used to present the key ideas behind the proposed RS-configuration paradigm and illustrate its advantages over the conventional resource allocation-based enforcement algorithm in §II-A. We then provide a general problem formulation and highlight the major objectives of the proposed paradigm in §II-B.

A. Network Model: Basic Notations and Illustration

Given a substrate network $G^S = (N^S, E^S)$, let N^S and E^S denote the set of substrate nodes and links, respectively. Considering a node $s \in N^S$, the total available resources at node s is defined as $\mathcal{R}_s$. Namely, any node s in N^S can allocate a maximum amount $\mathcal{R}_s$ of resources to VNFs. A VNF in a specific slice is not available and accessible by other network slices for the isolation purpose.

Let $\mathcal{G}^V$ be the set of RAN slices running over the substrate network G^S, where $\mathcal{G}^V = \{G^{V_1}, G^{V_2}, \ldots, G^{V_\ell}\}$. For emphasis, we will denote a single RAN slice as G^{V_i} $(1 \leq i \leq \ell)$ and drop i when the context is clear. We define a RAN slice $G^{V_i} = (N^{V_i}, E^{V_i}) \in \mathcal{G}^V$, where N^{V_i} and E^{V_i} are the set of VNFs and the set of virtual links between VNFs, respectively. For any VNF $u \in N^V$, it needs an amount $\Re_u$ of available resources at a substrate node to be embedded. For any virtual link $(u, v) \in E^V$, it requires a bandwidth $b_{(u,v)}$ for data flows between VNF u and VNF v. Likewise, instead of making the entire substrate network G^S available for routing traffic, we consider a more general case: we are restricted to the substrate network G^S with the capacity constraint matrix $\mathcal{C}$ for all substrate links $(s, t) \in E^S$, which specifies the maximum bandwidth $\mathcal{C}_{(s,t)}$ that can be allocated to all virtual links mapped onto (s, t). We introduce a binary mapping variable $\mathcal{M}_s^u$ to indicate the decision that the VNF u is mapped onto the substrate node s when $\mathcal{M}_s^u = 1$, and $\mathcal{M}_s^u = 0$ otherwise.

In the first place of setting up the network, all VNFs must be *mapped* (*embedded*) onto substrate nodes. To ensure that all VNFs will be embedded, the value of the binary mapping variable can be obtained as follows: $\sum_{s \in N^S} \mathcal{M}_s^u = 1$ for all $u \in \bigcup_{i=1}^{\ell} N^{V_i}$. However, due to the limited capacity at substrate nodes (available resources) and links (maximum bandwidth), a VNF u can actually be mapped onto a substrate node s if the available resources at s are sufficient ($\Re_u \leq \mathcal{R}_s$) and meet bandwidth and connectivity conditions such that $b_{(u,v)} \leq \mathcal{C}_{(s,t)}$, $(s,t) \in E^S$ for all $s, t \in N^S$, $v \in \mathcal{N}_u$, and $\phi_t^v = 1$, where $\mathcal{N}_u$ is the set of VNF u's neighbors. In addition, we use a variable $\psi_{(s,t)}^{(u,v)}$ to represent a simultaneous mapping of two VNFs u and v onto two substrate nodes s and t, respectively. In particular, $\mathcal{M}_s^u + \mathcal{M}_t^v - \psi_{(s,t)}^{(u,v)} \leq 1$[2].

Fig. 1(a) shows an example of VNFs embedded in substrate nodes of the RAN slicing. The substrate network consists of five base stations (BSs). Each BS provides four resource blocks (RBs) (two frequency units during two-time slots) to RAN slices. In the figure, there are also two RAN slices, and each consists of various VNFs. Both slices utilize the same substrate cellular network resources. We consider the VNF's resource requirement, as each needs a certain amount of resources at a substrate node to be embedded, where $R_{u_1} = 2$ RBs, $R_{u_1} = R_{p_2} = R_{p_3}$ and $R_{p_2} = 1$ RB (25% resources of a substrate node). In other words, if a VNF requires 25% of the spectrum resources, the substrate node should make 25% of the RBs to the VNF. In the figure, the solid link between any two VNFs in a slice indicates that they are neighbors (e.g., $(u_1, u_2), (p_1, p_2), (p_1, p_3)$ and (p_2, p_3)). Such a relation as above implies that for any VNF $u \in \bigcup_{i=1}^{\ell} N^{V_i}$, the target substrate nodes that the VNFs can be embedded in must be the substrate nodes OR one of the neighbors of the substrate nodes on which the VNFs neighbors are embedded. This connectivity constrain make u_1 cannot be embedded in the substrate node s_1 even though there are sufficient available resources.

As illustrated by the simple example above, existing works that focus on mapping VNFs by considering only the available resource of substrate nodes may fail to fully embed all VNFs. Moreover, they cannot leverage the maximum number of VNFs embedded, resulting in malfunction in RAN slices. Therefore, it is vital to come up with efficient algorithms to handle this resources allocation and embedding processes. The resources allocation algorithms will allow us to attain higher flexibility and embedding performance simultaneously,

[2]Such relations help avoid the quadratic constraint.

without needing further resources. To illustrate how resources allocation are handled in this case, consider again the simple example in Fig. 1(b) with a mapping plan to embed VNFs following a strict order. The result is that all VNFs are successfully embedded in the substrate network. We remark that the embedding algorithm should guarantee that all mapped VNFs meet the bandwidth and connectivity requirements.

B. RS-Configuration: Research Problem

Clearly embedding VNF strategy will not only hinge on substrate nodes resource allocation but also rely on the substrate connection set E^S of the substrate network and the bandwidth requirement of the virtual links between VNFs. This is a key research challenge we will tackle in this paper. In particular, for an RS-configuration in a cellular network, we would like to explicitly account for the *mapping plan* of VNFs, with the goal of providing a high performance for RAN slicing-based applications in terms of resources allocation for embedding VNFs in substrate node. A high performance in this stage is reflected through the number of successfully embedded VNFs. This is in contrast to many existing studies where no recoverable and stable performance is assured under failures for RAN slicing-based applications, nor is any attempt made to mitigate the impact of the RAN configuration process on the entire network. We observe that, in general, the enhanced embedding performance is achieved at the expense of *increased configuration units* the substrate network resources.

INSTANCE: Suppose we are given a substrate network $G^S = (N^S, E^S)$ with a set of substrate nodes and links, respectively. Each substrate node $s \in N^S$ can allocate to VNFs an amount $\mathcal{R}_s$ of resources. We consider a general case: we are restricted to the substrate network G^S with an *available* capacity constraint matrix $\mathcal{C}$ for all $(s,t) \in E^S$, which specifies the maximum bandwidth $\mathcal{C}_{(s,t)}$ that can be allocated to all virtual links mapped onto (s,t). Also, a set of RAN slices $\mathcal{G}^V = \{G^{V_1}, G^{V_2}, \ldots, G^{V_\ell}\}$ is running over the substrate network G^S. For any VNF, $u \in N^{V_i}$ requires an amount $\Re_u$ of available resources at a substrate node to be embedded. For any virtual link $(u,v) \in E^{V_i}$, it requires a bandwidth $b_{(u,v)}$ for data flows between two VNFs u and v.

QUESTION: Does a mapping plan ($\mathcal{MP}$) for all VNFs exist in the RAN slicing, such that the number of successfully embedded VNFs is not less than k? Mathematically, we formulate the following RS-configuration optimization problem:

$$\underset{\mathcal{M}_s^u}{\text{maximize}} \sum_{s \in N^S} \sum_{u \in \bigcup_i^{\ell} N^{V_i}} \mathcal{M}_s^u \tag{1}$$

Here, the set of variables $\mathcal{MP} = \{\mathcal{M}_s^u\}$ represents one possible mapping/embedding plan solution for VNFs in the RAN slicing. The objective function is to maximize the total number of mappings of VNFs, which means that the number of VNFs embedded is maximized. In addition, to ensure that one VNF can only be mapped onto at most one substrate node, they must meet the *mapping convergence constraint* at the first end s, $\sum_t \sum_v \psi_{(s,t)}^{(u,v)} = \mathcal{M}_s^u$, at the other end t, $\sum_s \sum_u \psi_{(s,t)}^{(u,v)} = \mathcal{M}_t^v$, and at both ends simultaneously, $\mathcal{M}_s^u + \mathcal{M}_t^v - \psi_{(s,t)}^{(u,v)} \leq 1$, for all $s,t \in N^S$, $u,v \in N^{V_i}$, and $\mathcal{M}_s^u$, $\mathcal{M}_t^v$, $\psi_{(s,t)}^{(u,v)} \in \{0,1\}$.

Apart from enhancing the embedding performance, we believe that latency is another key performance metric that must be accounted for in the design of the mapping plan for the RS-configuration. For example, delayed packets can cause the TCP timeout, resulting in unnecessary packet retransmissions, thus reducing overall application throughput. We, therefore, desire to *bound the bandwidth variability* in the mappings of VNFs as follows: $\sum_{u,v \in \mathcal{G}^V | \mathcal{M}_s^u = \mathcal{M}_t^v = 1} b_{(u,v)} \leq \mathcal{C}_{(s,t)}$ for all $(s,t) \in E^S$. Namely, the total allocated bandwidth on a substrate link does not exceed its link capacity. Likewise, the available resources at any substrate node s must be sufficient, that is, $\sum_{u \in \mathcal{G}^V | \mathcal{M}_s^u = 1} \Re_u \leq \mathcal{R}_s$ for all $s \in N^S$. On the other hand, to conserve the virtual connection between VNFs, the mapping must also meet the connectivity constraint in which for any mapping of a VNF u onto a substrate node s and the other VNF v onto the other substrate node t, for all $(u,v) \in E^V$, s and t must be physically connected in the substrate network, that is, $(s,t) \in E^S$. Thus, the mapping plan achieves the goal of a good embedding performance.

III. Resource Allocation Algorithms

The objective function in the equation 1 lays out the RS-configuration optimization problem thereby giving a better understanding of the design of the resource allocation algorithms which achieve the given optimization. In the following sections, the Group-Connectivity-based Algorithm (GCBA) and Group-based Algorithm (GBA) seek to maximize the embedding performance without being subject to a specific time constraint function. As part of each algorithm subsection, an overview and pseudocode are provided, along with a detailed description of the key idea behind the designed algorithms for the orderings and metrics involved.

A. Group-Connectivity-based Algorithm (GCBA)

Given a substrate network $G^S = (N^S, E^S)$, the major challenge becomes *how to select available resources and allocate them to suitable VNFs to achieve the highest number of embedded VNFs*. In this section we propose a high performance algorithm, referred to as Group-Connectivity-based Algorithm (GCBA) to attain the highest number of embedded VNFs, without being subject to the time complexity constraint. GCBA achieves a good performance by considering VNFs in a specific list of clusters, namely $\mathcal{L}^V$. We then find the best matching substrate group for a cluster set $\mathcal{N}_{u_i^+}$ $\forall$ $\mathcal{N}_{u_i^+} \in \mathcal{L}^V$.

To embed the VNFs in a given cluster set ($\mathcal{N}_{u_i^+}$) onto the appropriate substrate node, GCBA performs the three step $\mathcal{MP}$ set generation process as follows:

1) Firstly, we construct the descending-order list of VNF cluster sets $\mathcal{L}^V$ based on the size of each cluster set.
2) We then construct the ordered-set of substrate nodes $\mathcal{L}^S$ per VNF u such that $\mathcal{R}_t \geq \Re_u$ $\forall$ $t \in \mathcal{L}^S$ and $u \in \mathcal{L}^V$.

Algorithm 1 Group-Connectivity-Based Algorithm

Input: $G^S, \mathcal{G}^V$
Output: $\mathcal{MP}$

1: Initializing an empty set of mappings: $\mathcal{MP} = \emptyset$.
2: Constructing a descending-order (based on the size of each set) list of clusters $\mathcal{L}^V$, where $\mathcal{L}^V = \{\mathcal{N}_{u_i^+}, \dots, \mathcal{N}_{u_j^+}\}$ such that $\mathcal{N}_{u_i^+} \bigcap \mathcal{N}_{u_j^+} = \emptyset$.
3: **while** $\mathcal{L}^V \neq \emptyset$ **do**
4:     Let $\mathcal{N}_{u_i^+}$ be the first element in $\mathcal{L}^V$
5:     **for** $u \in \mathcal{N}_{u_i^+}$ **do**
6:         **EmbeddingGroup**$(u, \mathcal{MP})$
7:     **end for**
8:     $\mathcal{L}^V = \mathcal{L}^V - \{\mathcal{N}_{u_i^+}\}$
9: **end while**
10: return $\mathcal{MP}$;

3) Finally, we embed the mutually exclusive cluster sets in $\mathcal{L}^V$ to the substrate network.

As specified in algorithm 1, we construct the list of VNF cluster sets in line 2. This process is carried out by the iterative addition of the VNF cluster head v followed by its respective neighbor set $\mathcal{N}_v$. The representation of the set $\mathcal{L}^V$ is expressed by the following mathematical relation:

$$\mathcal{L}^V = [\{v_i \bigcup v_i'\}, \{v_j \bigcup v_j'\}, \dots, \{v_k \bigcup v_k'\}] \tag{2}$$

where $|\mathcal{N}_{v_i}| \geq |\mathcal{N}_{v_j}| \geq |\mathcal{N}_{v_k}| \; \forall \; v_i' \in \mathcal{N}_{v_i}, v_j' \in \mathcal{N}_{v_j}$, and $v_k' \in \mathcal{N}_{v+k}$ We construct the set of substrate nodes $\mathcal{L}^S$, containing the substrate nodes which satisfy the resource constraint $\mathcal{R}_t \geq \Re_u \; \forall \; t \in N^S$ and $u \in \mathcal{L}^V$. To optimize the embedding performance, we define the *neighborhood resource cumulative property* $x(T)$ for all the VNFs and substrate nodes so as to improve the selection process of the best possible substrate node. Given the VNF v and the substrate node $s \; \forall \; v \in \mathcal{L}^V$ and $s \in \mathcal{L}^S$, the $x(T)$ property for v and s is computed as follows:

$$\begin{aligned} v(T_{\Re}) &= \Re_v + \sum_{v' \in \mathcal{N}_v} \Re_{v'}, \\ s(T_{\mathcal{R}}) &= \mathcal{R}_s + \sum_{s' \in \mathcal{N}_s} \mathcal{R}_{s'} \end{aligned} \tag{3}$$

In equation 3, the replacement of x in $x(T)$ with v and s along with addition of $\Re$ and $\mathcal{R}$ as subscripts to T, allows for greater clarity while determining which element the $x(T)$ property is being computed for. In procedure Embedding-Group, we evaluate the $x(T)$ property (lines 3-4) after the embedding process is separated into two cases. For each case, the substrate node is picked based on $x(T)$.

Specifically, we first compute the difference $s(T_{\mathcal{R}}) - v(T_{\Re})$ (lines 6 and 11), after that the substrate node s, with the positive difference value closest to 0 is selected. In the case of only negative differences, the substrate node with the smallest negative difference is selected for embedding. The idea behind this metric (computing the difference) is to find the substrate node s with the highest probability of supporting the VNF u as well as the neighbor set $\mathcal{N}_u$. We select the substrate node with the difference value closest to 0 to minimize the possibility of the available resources being wasted. In addition, prioritizing the positive difference ensures maximum embedding (based on the resource constraint) of all the VNFs u' in $\mathcal{N}_{u^+}$, for all $\mathcal{N}_{u^+} \in \mathcal{L}^V$. Finally, if $\mathcal{L}^V$ consists of substrate nodes producing solely negative differences, the substrate node with the smallest difference is chosen so as to increase the probability of the maximum number of VNFs being embedded. The three steps to obtain the optimal $\mathcal{MP}$ set are therefore completed, thereby determining $\mathcal{MP}$.

B. Group-based Algorithm (GBA)

Algorithm 2 Group-Based Algorithm

Input: $G^S, \mathcal{G}^V$
Output: $\mathcal{MP}$

1: Initializing an empty set of mappings: $\mathcal{MP} = \emptyset$.
2: Evaluate the value $v(T_{\Re})$ for all VNFs v in N^{V_i}
3: Constructing a descending-order (based on the value of $v(T_{\Re})$) list of clusters $\mathcal{L}^V$, where $\mathcal{L}^V = \{\mathcal{N}_{u_i^+}, \dots, \mathcal{N}_{u_j^+}\}$ such that $\mathcal{N}_{u_i^+} \bigcap \mathcal{N}_{u_j^+} = \emptyset$.
4: **while** $\mathcal{L}^V \neq \emptyset$ **do**
5:     Let $\mathcal{N}_{u_i^+}$ be the first element in $\mathcal{L}^V$
6:     **for** $u \in \mathcal{N}_{u_i^+}$ **do**
7:         **EmbeddingGroup**$(u, \mathcal{MP})$
8:     **end for**
9:     $\mathcal{L}^V = \mathcal{L}^V - \{\mathcal{N}_{u_i^+}\}$
10: **end while**
11: return $\mathcal{MP}$;

While the set $\mathcal{MP}$ is obtained by the GCBA, we consider a second algorithm, Group-based Algorithm (GBA) that can be done with the similar computational complexity as well as provide a better embedding performance. The essential idea behind GBA is to order the VNF cluster sets based on the *neighborhood resource cumulative property* $(x(T))$. Specifically, to achieve the enhanced optimization, we change the basis of ordering the set of VNFs $\mathcal{L}^V$. Instead of ordering the VNF cluster sets based on the degree of cluster heads, descending-order cluster sets are constructed, based on the the neighborhood resource cumulative value $(x(T))$ of the cluster heads. In doing so, the cluster sets with the greatest isolation of resources are embedded onto the substrate layer first.

As shown in the algorithm 2, the step one (line 2) shows the improvement of GBA comparing to GCBA. Steps two and three, similar to the algorithm 1, are carried out with the help of the procedure EmbeddingGroup. The following steps are strictly followed the three steps of the embedding process. Hence, the $\mathcal{MP}$ set is obtained.

IV. Performance Evaluation

In this section, we present simulations to demonstrate our approaches' efficiency by firstly implementing the proposed algorithms for constructing a embedding plan for RS-

procedure EMBEDDINGGROUP(u, $\mathcal{MP}$)
Constructing a set $\mathcal{L}^S$ of substrate nodes such that $\mathcal{R}_{t'} \geq \Re_u \, \forall t' \in N^S$.
Evaluate the value $s(T_{\mathcal{R}})$ for all substrate nodes s in $\mathcal{L}^S$
Evaluate the value $u(T_{\Re})$
if $\mathcal{M}_{t'}^{v'} = 0 \; \forall \; v' \in \mathcal{N}_u, t' \in N^S$ **then**
Let t be a substrate node in $\mathcal{L}^S$ with the minimum difference $s(T_{\mathcal{R}}) - u(T_{\Re}) \; \forall \; s \in \mathcal{L}^S$
$\mathcal{R}_t = \mathcal{R}_t - \Re_u$
$\mathcal{MP} = \mathcal{MP} \bigcup \{\mathcal{M}_t^u\}$
else
Constructing $\mathcal{L}_{S^+}^{\mathcal{N}_u}$, representing the set of substrate nodes in $\mathcal{L}^S$ and their common neighbors, onto which the substrate nodes are hosting $v' \; \forall \; v' \in \mathcal{N}_u$
Let t_i be a substrate node in $\mathcal{L}_{S^+}^{\mathcal{N}_u}$ with the minimum difference $s''(T_{\mathcal{R}}) - v(T_{\Re})$ and $b_{(u,v')} \leq \mathcal{C}_{(t_i,t_j)} \; \forall \; s'' \in \mathcal{L}_{S^+}^{\mathcal{N}_u}, v \in \mathcal{N}_{u^+}, v' \in \mathcal{N}_u, \; t_j \in \mathcal{N}_{t_i^+}$, where $\mathcal{N}_{t_i^+}$ denotes the set of t_i's neighbors and itself.
if $\exists \; t_i$ **then**
$\mathcal{R}_{t_i} = \mathcal{R}_{t_i} - \Re_u$
$\mathcal{MP} = \mathcal{MP} \bigcup \{\mathcal{M}_{t_i}^u\}$
end if
end if
return $\mathcal{MP}$;
end procedure

configuration through simulations. We consider the RAN slicing consisting of one substrate network and a variety of network slices. It is feasible to create different embedding scenarios in the simulations and validate the simulation results. The first case is to test the feasibility of the proposed algorithms under the normal network condition that can provide reasonable resources for embedding VNFs into the substrate network, referred to as the **normal case**. In this case, we generate the number of substrate nodes from the interval $[60, 100]$, each is initialized with an amount of resources selected from the interval $[4, 8]$. The number of network slices is randomly generated from the interval $[2, 10]$ and the number of VNFs on each slice is also randomly chosen based on the number of network slices from the interval $[10, 100]$.

In the second case, we generate the network topology with limited resources that can test the performance of the proposed algorithms in the resource shortage condition, referred to as the **shortage case**. In this case, network resources are strictly controlled as follows: for the number of substrate nodes, from the interval $[60, 100]$; for the number of resources for each substrate node, from the interval $[2, 4]$; for the number of network slices, from the interval $[2, 10]$; and for the number of VNFs on each slice, from the interval $[1, 10]$. In the following sections, Fig. 2 shows results in the "normal case" test and Fig. 3 shows results in the "shortage case" test. To ensure the performance of the proposed algorithms are fairly verified, we compare them with the other two approaches, which consider

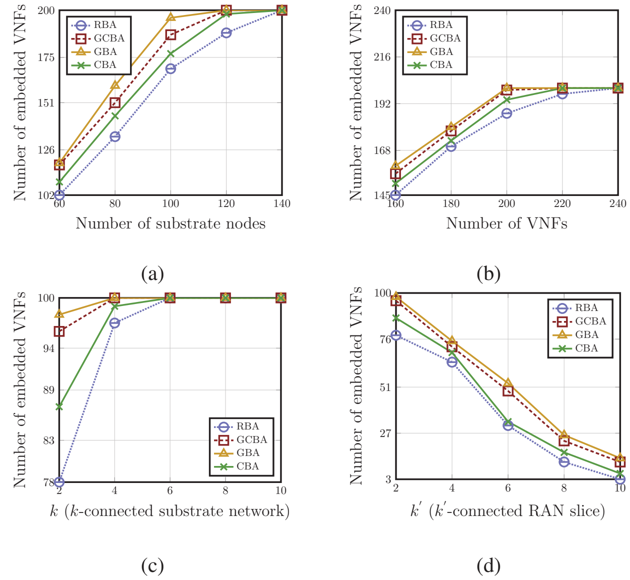


Fig. 2: **Normal Case:** total number of embedded VNFs when varying a) the number of substrate nodes, b) number of VNFs, c) k, and d) k'.

the resources and connectivity of VNFs as the primary factor for the embedding process, referred to as Resource-based Algorithm (RBA) and Connectivity-based Algorithm (CBA).

A. Embedding Performance

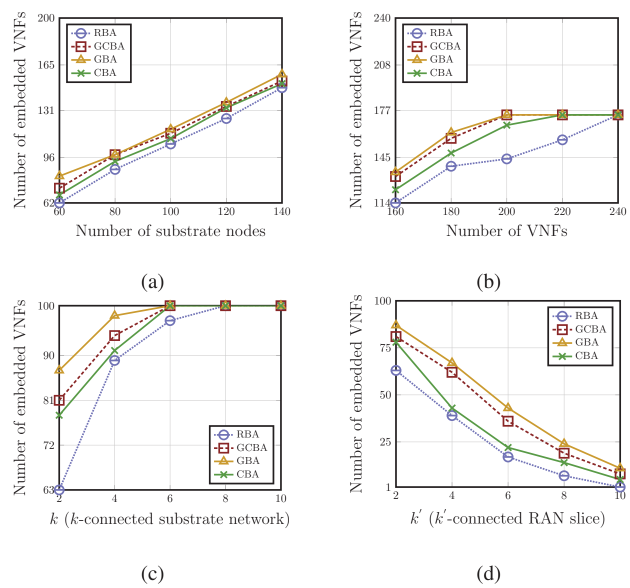


Fig. 3: **Shortage Case:** total number of embedded VNFs when varying a) the number of substrate nodes, b) number of VNFs, c) k, and d) k'.

Results in Fig. 2 show the performance of all algorithms in the "normal case" test. We test the proposed algorithms using different metrics, including varying the number of substrate nodes, changing the number of VNFs, adjusting the degree of substrate nodes (k) and the degree of VNF (k'). Specifically, Fig. 2(a) shows comparisons of the number of embedded VNFs when the number of substrate nodes ranges from 60 to 140. In the figure, GBA and GCBA provide the best performance with a higher total number of embedded VNFs. During the embedding process, GBA and GCBA consider not only the required resources of a single VNF but also we take into account of the required resources of a cluster of VNFs. In addition, in the GCBA, we also consider the degree of VNFs (number of neighbors) when determining the mapping plan. The concept of cluster (interdependency) makes the GBA and GCBA to be the best in performing embedding VNFs. The differences are reflected through the *neighborhood resource cumulative property* $x(T)$ considered in both GBA and GCBA.

In Fig. 2(b) we test the performance of the proposed algorithms when the number of VNFs ranging from 160 to 240. We realize that the higher the VNFs in the substrate network, the higher the number of VNFs is successfully embedded into the substrate network. It indicates that the algorithms perform more efficiently when having more resources in the network (higher number of VNFs). Likewise, in Fig. 2(c) and Fig. 2(d) we analyze comparisons of the capability of the proposed algorithms when varying the value of k and k', where k and k' denote the degree of every substrate node in the substrate network and the degree of VNFs in RAN slices, respectively. In these figures, GBA and GCBA demonstrate a better performance with a higher total number of embedded VNFs when comparing to CBA and RBA. We also observe that the higher k or k', the higher the number of VNFs embedded in the substrate network. In addition, there is higher efficiency in performance when k or k' is increased.

In Fig. 3 we test the performance of the proposed algorithms in the "shortage case" test to see how the shortage resource condition impacts to the number of VNF embedded in the network. In Fig. 3(a) we compare the embedding performance of the proposed algorithms to that the number of substrate nodes ranging from 60 to 140. As shown in the figure, GBA and GCBA have the best efficiency performance with a higher total number of embedded VNFs. As mentioned in the previous analysis, both GBA and GCBA consider the *neighborhood resource cumulative property* $x(T)$. In addition, in GBA we evaluate the $x(T)$ for both virtual networks and substrate network while only the substrate network is evaluated with the $x(T)$ in GCBA. These metrics make GBA and GCBA perform better than CBA and RBA. In contrast, using CBA and RBA, the embedding performance is lower with a lower number of embedded VNFs. Note that for CBA and RBA, all VNFs are scheduled to be embedded by considering only the embedding status of the individual VNFs resource or connectivity constraint. Likewise, in Fig. 3(b)–3(d) we test to see how the number of VNFs, k, and k' impact the number of successfully embedded VNFs. GBA and GCBA demonstrate a greater efficiency performance with a higher number of VNFs embedded in the substrate network whereas, RBA and CBA have a lower total number of VNFs embedded in the substrate network. We also observe that the greater k or k', the better the number of VNFs embedded in the substrate network as well. In contrast, RBA and CBA have a fewer total number of embedded VNFs as they have a lower embedding efficiency.

V. Conclusion

In this paper we have established the theoretical foundation for using RS-configuration to construct a VNF embedding plan for the RAN slicing. We have formulated the efficient resource allocation as an essential problem with the objective to maximize the total number of embedded VNFs. To solve the RS-configuration problem, we have introduced two efficient algorithms (GCBA and GBA) and show theoretical analyses to demonstrate the efficacy of the algorithms. Extensive simulation results have been provided to evaluate the performance of the proposed algorithms using different metrics in terms of the embedding performance and resource utilization performance. We have created different scenarios to test the algorithms, including considering the network under the normal condition (normal case) and under the resource shortage condition (shortage case). Through the results of simulations, GBA and GCBA consistently show a better performance for embedding VNFs in comparison to the other two approaches RBA and CBA within reasonable bounds of computational complexity.

Acknowledgement

This research was supported in part by US NSF grants CNS-2103405 and CNS-1814614.

References

[1] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "How should i slice my network? a multi-service empirical evaluation of resource sharing efficiency," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18, 2018, p. 191–206.

[2] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5g ran slicing for verticals: Enablers and challenges," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 28–34, 2019.

[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.

[4] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for ran network slicing," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 970–973, 2018.

[5] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "Nfp: Enabling network function parallelism in nfv," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17, 2017, p. 43–56.

[6] D. Mascarenas, E. Flynn, C. Farrar, G. Park, and M. Todd, "A mobile host approach for wireless powering and interrogation of structural health monitoring sensor networks," *IEEE Sensors Journal*, vol. 9, no. 12, pp. 1719–1726, Dec 2009.

[7] H. Halabian, "Optimal distributed resource allocation in 5g virtualized networks," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 28–35.

[8] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5g systems," *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, 2019.

[9] S. D'Oro, F. Restuccia, and T. Melodia, "Toward operator-to-waveform 5g radio access network slicing," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 18–23, 2020.

[10] S. Schneider, M. Peuster, and H. Karl, "A generic emulation framework for reusing and evaluating vnf placement algorithms," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2018, pp. 1–6.