An Efficient Mapping Algorithm with Novel Node-Ranking Approach for Embedding Virtual Networks

Haotong Cao[1,3], Yongxu Zhu[2], Gan Zheng[2] and Longxiang Yang[1,3]

1. College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China
2. Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Leicestershire, UK
3. Key Laboratory of Broadband Wireless Communication and Sensor Network Technique, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing, China

***Abstract*—Virtual Network Embedding (VNE) problem has been widely accepted as an important aspect in Network Virtualization (NV) area: how to efficiently embed virtual networks, with node and link resource demands, onto the shared substrate network that has finite network resources. Previous VNE heuristic algorithms, only considering single network topology attribute and local resources of each node, may lead to inefficient resource utilization of the substrate network in the long term. To address this issue, a Topology Attribute and Global Resource-Driven VNE algorithm (*VNE-TAGRD*), adopting a novel node-ranking approach, is proposed in this paper. The novel node-ranking approach, developed from the well-known Google *PageRank* algorithm, considers three essential topology attributes and global network resources information before conducting the embedding of given virtual network request (VNR). Numerical simulation results reveal that the *VNE-TAGRD* algorithm outperforms five typical and latest heuristic algorithms that only consider single network topology attribute and local resources of each node, such as long-term average VNR acceptance ratio and average revenue to cost ratio.**

Index Terms—Virtual Network Embedding; topology attribute; global resource; node-ranking approach; VNE-TAGRD

I. INTRODUCTION

In recent years, network virtualization (NV) has attracted intensive attention from both academic [1][2] and industry [3][4]. Current Internet is impeded to further develop due to the network ossification [2][3]. Therefore, network virtualization is widely accepted as the promising solution for the future network and also works as a key enabler for the cloud computing [5]. NV is able to provision many isolated virtual networks (VNs) coexist on a substrate network (SN) for further sharing the physical computing and networking resources simultaneously and seamlessly.

In general, the VN is a logical topology consisting of a set of virtual nodes (e.g. virtual routers) interconnected by corresponding virtual links. To the shared substrate network, it consists of substrate nodes (e.g. routers, switches) connected by substrate links (coaxial cable or optical fiber) that form the substrate topology. In NV area, the universal business model [6], adopted in this paper, is that the conventional Internet service provider (ISP) is decoupled into infrastructure providers (InPs) and service providers (SPs). Infrastructure providers (InPs) are responsible for managing and running underlying network infrastructures. SPs can dynamically construct different VNs to fulfill the different demands of end-users by renting underlying resources from the InPs. Mapping or embedding given VNs, requested by end users, onto the shared substrate network is known as the so-called Virtual Network Embedding (VNE) in the literature. To different roles (e.g. InP, SP) in NV, the optimal embedding of each VN is different, with regard to each concrete goal. While in this paper, the minimization of VN embedding cost and maximization of VN acceptance ratio (for the InP) are taken into account. Survivability of each mapped VN (for different end-users and SPs) are not considered in this paper. In addition, the fundamental and important VNE network scenario (one SN and several given VNs in a continuous time event) is considered in this paper. Other complex network scenarios (e.g. occasional substrate node / link failure or multiple underlying substrate networks) can be extended on this fundamental basis for the future research.

VNE problem has proven to be a NP-hard problem [7] before. Detailed VNE surveys [8][9] has been conducted in the literature. Derived from surveys, VNE algorithms are mainly classified into two categories: the exact [9] and the heuristic [8]. The meta-heuristic [8] is not talked in our paper. Though VNE exact algorithms enable to ensure an optimal or near-optimal embedding of each given VN in small-scaled network scenarios [9], the computational complexity are very large in a discrete time event [10-12], not to mention medium or large-sized network scenario in a continuous time event. Other exact-like algorithms [13-15] have to relax integer constraints or use column generation approach to achieve a feasible VN embedding in limited time. Strictly speaking, these algorithms [13-15] belong to the heuristic category. Therefore, it is of great importance to develop heuristic algorithms. Many typical heuristic algorithms [16-22] have been proposed to solve VNE problem over the past years. These heuristic algorithms conduct the embedding of each given VN in two stages, node mapping and link mapping stage. However, these algorithms conduct the node mapping, only based on single topology attribute (e.g. node degree or node strength) or the local network resource (e.g. the nodes' capacity resources or the product of nodes' capacity resources and their adjacent link bandwidth). Another two latest heuristic algorithms [21][22] take global network resource into consideration in the node mapping stage. However, over-simplified global resource metrics (e.g. the global node capacity ratio) are used in [21] and [22]. In addition, the benefits of considering global resources are not fully explored in these algorithms, not to mention benefits of considering multiple topology attributes simultaneously in embedding VNs.

In this paper, a Topology Attribute and Global Resource-Driven mapping algorithm *VNE-TAGRD* is proposed to deal with VNE problem in a continuous time event. The goals of *VNE-TAGRD* are to help the InP to minimize VNR embedding cost (**Fig. 4**) and maximize VNR acceptance ratio (**Fig. 3**) in the long term. When embedding one VNR each time, the *VNE-TAGRD* adopts a novel node-ranking approach, similar to the *PageRank* approach in web-searching area [23-25], to rank all substrate nodes and virtual nodes ahead, according to three essential topology attributes and global network resources. The greedy node mapping approach, based on the novel node-ranking values, is then performed, with fulfilling the node constraints (virtual node location and virtual node capacity demands are considered in our paper). When the node mapping is accomplished, the following link mapping is implemented with the shortest-path (SP) algorithm [37] (virtual link bandwidth requirement and virtual link propagation delay demand are considered). To further prove the efficiency of *VNE-TAGRD*, a comprehensive simulation is also conducted in our paper. Numerical simulation results reveal that the *VNE-TAGRD* algorithm outperforms five representative and latest heuristic algorithms, considering single topology attribute and local network resource, in terms of long-term average virtual network request (VNR) acceptance ratio (**Fig. 3**) and average revenue to cost ratio (**Fig. 4**) in a continuous time event.

Main contributions of this paper are listed below:

1) A novel node-ranking approach (***Algorithm 1***), stimulating from the *PageRank* method [23-25], is propose by considering several essential topology attributes and global network resources [28-29]. The novel node-ranking approach is different form the previous universal node-ranking approaches [16-22] and ensures efficient substrate resource utilization.

2) A heuristic algorithm *VNE-TAGRD* is proposed based on the novel node-ranking approach. The number of node-link constraints considered in *VNE-TAGRD* is up to four (node location, node capacity, link bandwidth and link propagation delay) while the universal VNE algorithms only consider two (node capacity and link bandwidth). Apart from the node location requirement, link propagation delay, conducted in link mapping stage (***Algorithm 2***), has not been considered as a node-link constraint in previous VNE research area.

With the increasing of delay sensitive services [30-32] on the network, smaller and guaranteed transmission delay is needed for this kind of new services. We therefore consider the link propagation delay as a node-link constraint into *VNE-TAGRD* algorithm, so as to the mapped virtual network provides delay guaranteed services to meet future service requirements.

3) A comprehensive simulation is conducted to validate the efficiency and advantage of *VNE-TAGRD* algorithm. Five typical and state-of-the-art heuristic algorithms, closely related to our *VNE-TAGRD*, are selected to make up the comparison. Simulation results vividly show that the *VNE-TAGRD* algorithm outperforms the existing heuristic algorithms in the long term, such as average VNR acceptance ratio.

The rest of our paper is organized as follows. Related work is presented in Section II. Section III formulates the VNE problem, particular for the definition of VN and VNR in VNE research. In Section IV, details of algorithm *VNE-TAGRD* are presented. The simulation work is implemented in Section V, along with simulation parameters setting. At last, conclusion and future work are briefly talked in Section VI.

II. Related Work

VNE aims to realize the optimal node and link mapping of each given VNR simultaneously, and can be formulated as an unsplittable flow problem [37] that its solution is NP-hard [7]. Previous researches concentrate on evaluating different algorithms' abilities of batching processing given VNs in a discrete time window / time event (e.g. [10-11]) and finding out the optimal VN mapping of each given VN. However, it is hardly possible to embed proposed VNRs optically in a continuous time event. Therefore, VNE algorithms are mostly heuristic in the literature, compromising global optimality for a short execution time. Based on these backgrounds, this section only discusses VNE algorithms that are closely related to *VNE-TAGRD* algorithm. Any reader who has great interest in other extended VNE issues, such as multiple substrate networks [12][43], VN survivability [26][41][42] and evaluating algorithms in a discrete time event, can refer to [8-9] for detailed references.

A. Typical and Related Heuristic Algorithms

In reference [16], a typical heuristic algorithm, proposed by Yu *et al.*, applies the greedy method to implement the virtual node mapping in the first step. Dijkstra's algorithm and multi-commodity flow approach [37] are then used to deal with the link mapping. Before conducting the greedy node mapping, a node-ranking approach is used to rank all substrate nodes and virtual nodes. The node-ranking value of each node is simply determined by the product of node capacity and all its adjacent link bandwidth. The direct node-ranking value is enable to indicate the embedding ability of the corresponding node, to some degree. Virtual nodes are embedded onto the substrate nodes by processing the two sorted node lists. Therefore, the virtual node with the highest node-ranking value among the remaining ones of the VNR will always be embed onto the substrate node that also has the highest node-ranking value among the remaining substrate ones, whose available resource meets the VNR demands (node capacity is only considered in [16]). Yu *et al.* also introduces the concept of path splitting to accept more proposed VNRs and relieve substrate link loading. However, this algorithm will lead to a level of resource fragmentation that is unfeasible for accepting VNRs with large sizes. The reason for resource fragmentation is the inefficient node-ranking approach. In addition, only constraints of node capacity and link bandwidth are considered in [16]. Therefore, there is no restriction of the VNE solution searching space. If the size of SN becomes larger, the computational complexity will be larger.

Cheng *et al.* [17] proposes another topology-aware node mapping approach that uses the *Markov Random Walk model* to measure the node capacity and its joint link bandwidth. Stimulating from the *PageRanking* algorithm, steady state node-ranking values of a network is able to be calculated through a classic iterative scheme. The steady state node-ranking values are then adopted in the greedy node mapping. The virtual links, whether splittable or unsplittable, are then mapped to substrate network either using the k-shortest path or multi-commodity flow method. However, the VN embedding approach proposed in [17] still leads to the problem of resource fragmentation and inefficient substrate utilization in the long term, same to reference [16]. The cause of this problem lies to the fact that the node-ranking approach of [17] only considers one node capacity and all its adjacent link bandwidth. Other topology attributes and global network resources (e.g. node location) are still not considered in the node-ranking approach. References [21][22] conduct the similar node-ranking approach and leads to the same problem, as described in [17], too. In addition, Authors of [17] also use the same topology-aware node mapping approach to deal with the energy-aware VNE problem [18] while the VNE energy-aware problem has nothing to do with the scope of our paper.

In reference [19], Feng *et al*. introduces several different topology attributes and proposes three different node-ranking approaches. However, these three node-ranking methods are still simply determined by the product of node capacity, total link bandwidth and intermediate nodes, similar to reference [16]. Steady state node-ranking values of a substrate / virtual network are not calculated in [19]. All substrate nodes and virtual nodes of one VNR are then ranked in decreasing order, according to the direct product values. The virtual node with the highest node-ranking value among the remaining ones of the same VNR will always be embed onto the substrate node that also has the highest node-ranking value among the remaining substrate ones, whose available resource meets the VNR demands. Greedy method is adopted to implement the virtual node mapping in the following step. Dijkstra's algorithm is then used to deal with the link mapping. Path splitting is not considered in [19]. Authors of [19] do not explore the benefit of the global resource information in VNE, either. Therefore, this will lead to inefficient substrate resource usage in the long term. In addition, only node capacity and link bandwidth constraints are considered in embedding each given VNR [19], no highlighting the global resource information.

In reference [20], authors propose another heuristic VNE algorithm to deal with VNE problem. The main highlight of [20] is the proposed node-ranking approach. The node-ranking approach is based on the node degree and the clustering coefficient information. The technique of node importance metric is adopted in [20] to rank the substrate nodes, aiming to select the node with the most embedding potential for every virtual node in each VNR. The greedy node mapping follows after the node-ranking approach. In the link mapping stage, the k-shortest path is also adopted, same to the above references. However, reference [20] only considers the local resources of nodes and its neighborhood nodes, ignoring other topological attributes and global network resources, and leads to lower resource utilization of the substrate network in the long run. Numerical simulations results (Section V) prove this true.

B. Representative and Related Exact VNE Algorithms

With the development of VNE algorithms, some researchers also turn to studying the exact algorithm [9] to find the optimal embedding per VN, with respect to a concrete object. The VNE problem is NP-complete in several VNE cases. Software tools (e.g. GLPK [44], CPLEX [45]), suitable model (e.g. relaxed LP model, restricted ILP model) [46] and small-scaled network instances all contribute to solving each VN embedding in reasonable time. VNE exact algorithm is able to offer a benchmark for performance comparison with other heuristic algorithms. This *subsection* talks about some representative exact and exact-like algorithms in the literature.

The algorithm, proposed by reference [13] is widely considered as a representative exact-like algorithm in VNE research area. Chowdhury *et al.* firstly solves the VNE problem by applying the mixed integer programming (MIP) model of the optimization theory [31]. Due to the complexity of using MIP to directly solve a medium-sized network, Chowdhury *et al.* has to modify the MIP model into a linear programming (LP) model and relaxes the integer constraints. Chowdhury *et al.* uses the LP model to solve the virtual node mapping. If the solution of LP model is feasible, the virtual links are assigned to corresponding substrate paths by using Dijkstra's algorithm [37] or multi-commodity flow approach. Though the authors coordinate node and link mapping, reference [13] is not an efficient and effective VNE algorithm, in terms of average VNR acceptance ratio in the long term. In addition, node-link constraints, conducted in the VN embedding, are node capacity and link bandwidth.

Houdi *et al.* [12] proposes an exact algorithm to map one VN across multiple SNs, fighting for the minimization of VN provision cost. However, the cost metric is not defined clearly in [12]. The authors use the *max-flow and min-cut method* to split one VN into sub-VNs firstly. The number of parts is equal to the number of SNs. Then the LP model is adopted to map each part of the VN to each SN. Though splitting a VN reduces the complexity, [12] restricts the solution space and can not find the feasible embedding of the VN in many cases. The topology and resource information of all substrate networks, unknown in practical networking environment, are assumed known in [12]. Thus, multiple SNs can be seen as a large substrate network.

Betero *et al.* [10], a variation of reference [13], adds the constraint of allowing more than one virtual node per VN to map onto the same substrate node and uses the pure MIP model [13] to deal with VNE problem directly. The goals of [10] are to minimize the consumption of link bandwidth and the number of active substrate nodes. This paper is also the first attempt in VNE energy aware problem. Only node capacity and link bandwidth are taken into account in the simulation part. Due to the computational complexity of MIP, the sizes of SN (15 nodes) and VNRs are set small.

While in reference [11], presented by Melo *et al.*, a pure ILP model is proposed to solve VNE problem in a discrete time event. The aims of [11] are to minimize the VNR cost and achieve substrate load balancing. Two node-link constraints, node location requirement and virtual link propagation delay, are mentioned. However, both two constraints are not analyzed. Virtual link propagation delay constraint and node location constraint are not included in the simulation work, either. Generally speaking, reference [11] is an exact algorithm and fights for the substrate load balancing, only considering node capacity and link bandwidth constraints, in the fundamental network scenario.

Mijumbi *et al.* [15], stimulating from [14], is another variation of [13] to solve the VNE problem. Constraints considered are same to what are considered in [13]. In the first step, [15] formulates the VNE problem by using the pure MIP model [13]. Relaxing the binary variables to take on continuous values, the MIP model is relaxed into LP model. With solving the LP model and getting the node mapping done, the shortest path is selected again [37]. These procedures are same to what are described in [13]. The authors then derive the corresponding dual MIP model. The dual model is used to ensure the selected paths legitimate. The VN assignment which consumes less substrate resources is preferred. The time complexity decreases a lot, comparing with the pure MIP model. This exact algorithm also improves the VNR acceptance ratio in the long term, comparing with the heuristics [13].

C. Brief Summary

To summarize, because of VNE problem NP-hard, previous VNE algorithms in the literature are mostly heuristic. Remaining algorithms are exact and usually solve VNE problem by using the optimization theory approach [46]. However, exact algorithms (***subsection B***) usually have large computational complexity and cannot be promoted to embedding VNRs in a continuous time event, comparing with the heuristics (***subsection A***). Therefore, it is important and necessary to develop VNE heuristic algorithms. The heuristics solve the VNE problem in two separate stages, using greedy method or relaxed LP model in node mapping stage and optimizing the link mapping by Dijkstra's algorithm [37] or multi-commodity flow approach. Therefore, it is important to propose an efficient node-ranking approach before conducting node mapping. In addition, only node capacity and link bandwidth requirements are considered to embed a VNR in previous VNE algorithms. Other node-link constraints (e.g. virtual link propagation delay, virtual node location demand) are usually not taken into account.

The *VNE-TAGRD* algorithm, proposed in this paper, differs from previous heuristic algorithms in many aspects. First of all, the algorithm *VNE-TAGRD* adopts a novel node-ranking approach, stimulating from the well-known *PageRank* method in web-searching area, to rank all substrate nodes and virtual nodes before conducting the embedding of each given VNR. The novel node-ranking approach takes three essential topology attributes and global network resources into consideration. Previous heuristic algorithms only consider one topology attribute (e.g. node degree or node strength) and local network resource (e.g. the product of one node capacity and sum of node's all adjacent link bandwidth) to rank all nodes of the given network. Therefore, these universal node-ranking approaches [16-22] cannot ensure efficient substrate resources utilization in the long term. Then, the *VNE-TAGRD* coordinates the node and link mapping to achieve a better embedding of each VNR, comparing with the heuristics. The node mapping is an important stage in VNE since it determines the efficiency of the link mapping. Coordinating and designing effective node-ranking approach will contribute to better resource utilization [13] and efficient link mapping. Next, apart from the universal constraints of node capacity and link bandwidth, virtual node location and virtual link propagation delay are also added as node-link constraints to be considered in running the *VNE-TAGRD*. Particularly, the virtual link propagation delay is considered as a constraint for the first time in VNE area. With the increasing of delay sensitive services on the network, guaranteed transmission delay is needed. The propagation delay constraint in *VNE-TAGRD* aims to offer delay guaranteed VNR to meet future service requirements [30-32]. Finally, authors of this paper implement a comprehensive simulation against the typical and related heuristic algorithms. The efficiency and advantage of *VNE-TAGRD* are validated in Section V.

III. VNE Problem Description

This section presents a generic description of VNE problem, including substrate and virtual network models, VN embedding procedures and fundamental evaluation metrics.

A. VNE Network Model

1) *Substrate network*: The substrate network in VNE is modeled as a weighted graph $G^S=(N^S, L^S)$, where N^S is the set of all substrate nodes (e.g. routers or switches) and L^S is the set of all substrate links (e.g. coaxial cable). Each substrate node $\boldsymbol{m} \in N^S$ is characterized by its node capacity, $\mathbf{C}^S_m$, and the node location, loc($\boldsymbol{m}$). The location of substrate node $\boldsymbol{m}$, loc($\boldsymbol{m}$), is defined by $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates in this paper. With respect to substrate links, each substrate link $\boldsymbol{mn}$ has a finite bandwidth $\mathbf{B}^S_{mn}$ and a substrate link propagation delay $\mathbf{D}^S_{mn}$. The set of all loop-free substrate paths is denoted by notation $\mathbf{P}^S$. $\mathbf{P}^S_{mn}$ is the set of all loop-free paths between substrate nodes $\boldsymbol{m}$ and $\boldsymbol{n}$. $\mathbf{P}_{mn}$ is one path selected from the $\boldsymbol{mn}$ path set $\mathbf{P}^S_{mn}$. The right part of Fig. 1 shows one substrate network. The numbers over the links are available link bandwidth and numbers in rectangles are available node capacity. For simplicity, the location of each substrate node and link propagation delay of each substrate link are omitted in this figure.

2) *Virtual network request:* In VNE research area, each virtual network also can be modeled as a weighted graph $G^V=(N^V, L^V)$, where N^V is the set of all virtual nodes and L^V is

the set of all virtual links. Each virtual node $\boldsymbol{M}$ is characterized by the required node capacity, $\mathbf{C}_{M}^{\mathrm{V}}$, and its required virtual node location, Loc($\boldsymbol{M}$). The allowed maximum deviation of virtual node $\boldsymbol{M}$ is LR($\boldsymbol{M}$). The deviation of one virtual node $\boldsymbol{M}$ and one of its potential candidate substrate node $\boldsymbol{m}$ must be within the maximum deviation LR($\boldsymbol{M}$) of the virtual node $\boldsymbol{M}$. With respect to virtual links, each virtual link $\boldsymbol{MN}$ has a required bandwidth $\mathbf{B}_{MN}^{\mathrm{V}}$ and required virtual link propagation delay $\mathbf{D}_{MN}^{\mathrm{V}}$. With adding the time attributes (e.g. maximum waiting time, arriving time, duration time, leaving time), the VN is extended to be a virtual network request (VNR). The left part of Fig. 1 shows two virtual networks requests with different topologies. The numbers over the links are required link bandwidth and the numbers in rectangles are node capacity demand. For simplicity, all nodes' location requirements, required virtual links' propagation delay and time attributes of each VNR are not plotted either.

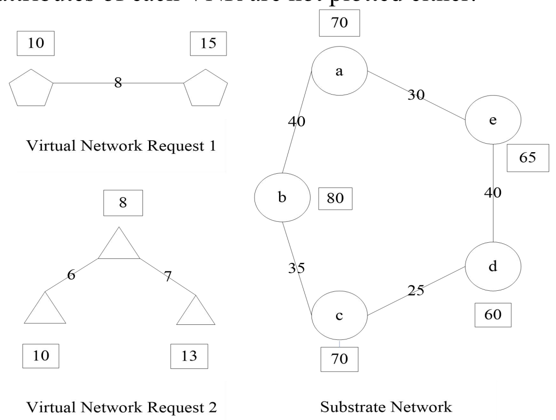

Fig. 1. One Substrate Network and Two Virtual Network Requests

B. VNR Embedding and Evaluation Metrics

The generic embedding of each VNR consists of two main stages: the stage dealing with the embedding of virtual nodes of the VNR, and the stage ensuring the embedding of virtual links of the VNR.

1) *Node Embedding Stage*: To each VNR, each virtual node must be assigned to a different substrate node of the SN. This does benefit to the flexibility and manageability of SN. The assignments of all virtual nodes in one VNR are determined by the node-mapping function $\boldsymbol{F}_{\mathrm{N}}(\ ): \mathrm{N}^{\mathrm{V}} \rightarrow \mathrm{N}^{\mathrm{S}}$.

$$\boldsymbol{F}_{\mathrm{N}}(\boldsymbol{M}) \in \mathrm{N}^{\mathrm{S}}$$

$$\boldsymbol{F}_{\mathrm{N}}(\boldsymbol{M}) \neq \boldsymbol{F}_{\mathrm{N}}(\boldsymbol{N}), \quad \text{if and only if } \boldsymbol{M}=\boldsymbol{N}$$

subject to

$$\mathbf{C}_{M}^{\mathrm{V}} \leq \mathbf{R}^{\mathrm{S}}(\boldsymbol{F}_{\mathrm{N}}(\boldsymbol{M})) \tag{1}$$

$$\mathrm{Dis}(\mathrm{loc}(\boldsymbol{F}_{\mathrm{N}}(\boldsymbol{M})), \mathrm{Loc}(\boldsymbol{M})) \leq \mathrm{LR}(\boldsymbol{M}) \tag{2}$$

where Formula 1 aims to ensure the node capacity demand of virtual node $\boldsymbol{M}$ must not exceed the available capacity of the selected substrate node so as to accommodate virtual node $\boldsymbol{M}$; Formula 2 aims to ensure that the deviation relationship between the virtual node $\boldsymbol{M}$ and the selected substrate node must be within the required radius LR($\boldsymbol{M}$). $\mathbf{R}^{\mathrm{S}}(\ )$ represents the available node capacity in Formula 1. Both two formulas must be fulfilled simultaneously in the node embedding stage.

2) *Link Embedding Stage*: Each virtual link of the same VNR is mapped onto a single substrate path in this paper between the corresponding substrate nodes that host two end virtual nodes. In this paper, path splitting [26] cases are not considered. The link embedding is performed according to a link-mapping function $\boldsymbol{F}_{\mathrm{L}}(\ ): \mathrm{L}^{\mathrm{V}} \rightarrow \mathrm{P}^{\mathrm{S}}$ for all virtual links per VNR.

$$\boldsymbol{F}_{\mathrm{L}}(\boldsymbol{M}, \boldsymbol{N}) \subseteq \mathrm{P}^{\mathrm{S}}(\boldsymbol{F}_{\mathrm{N}}(\boldsymbol{M}), \boldsymbol{F}_{\mathrm{N}}(\boldsymbol{N}))$$

subject to

$$\mathbf{B}_{MN}^{\mathrm{V}} \leq \mathbf{R}^{\mathrm{S}}(\mathrm{P}_{F_N(M)F_N(N)}) \tag{3}$$

$$\mathbf{D}_{MN}^{\mathrm{V}} \leq \mathbf{D}^{\mathrm{S}}(\mathrm{P}_{F_N(M)F_N(N)}) \tag{4}$$

where Formula 3 aims to ensure the link bandwidth demand of any virtual link $\boldsymbol{MN}$ must not exceed the available bandwidth of one selected substrate path that accommodates $\boldsymbol{MN}$. $\mathbf{R}^{\mathrm{S}}(\ )$ represents the available bandwidth of one selected substrate path in Formula 3. Formula 4 aims to ensure the substrate path propagation delay of one selected substrate path must not exceed the required link propagation delay of the virtual link $\boldsymbol{MN}$ $\mathbf{D}_{MN}^{\mathrm{V}}$. Both two formulas must be fulfilled simultaneously in the link embedding stage. In the VNR embedding process, the node and link embedding must be fulfilled at the same time, too.

For better understanding one VNR embedding in a continuous time event, the embedding process of one VNR is depicted in **Fig. 2**.

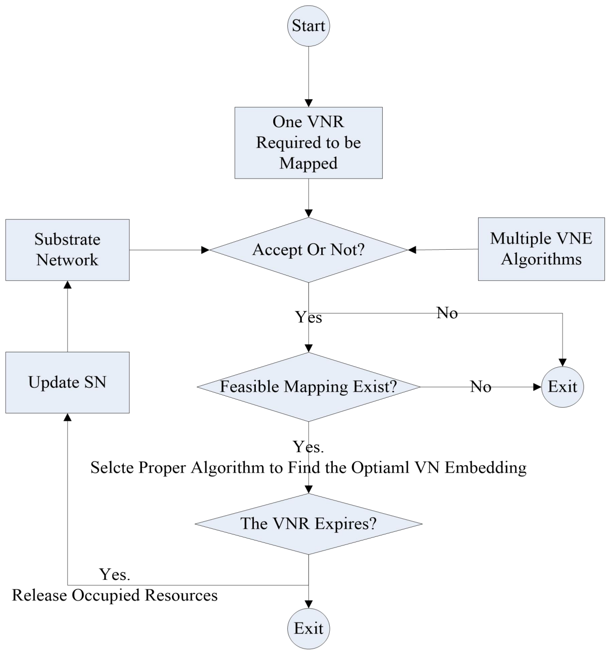

Fig. 2. Diagram of Embedding of One VNR

With successfully embedding one VNR, it is essential to define corresponding metrics so as to evaluate the selected embedding algorithm.

In this paper, a “pay-per-user” revenue model, on the basis of the “on-demand” cloud service price scheme by Amazon Web Services (AWS) [27] is adopted to calculate the revenue of each accepted VNR. The revenue of a VNR in a continuous time event is given by follows.

$$\boldsymbol{Rev}(G_T^V)=\begin{cases}\boldsymbol{Rev}(G^V)\bullet T, & \text{if the } \boldsymbol{G}^V \text{ is accepted}\\ 0, & \text{else}\end{cases} \tag{5}$$

In Formula 5, T represents the duration time of the VNR G^V. Similar to other references [16]-[22], the per-unit revenue of a G^V ($\boldsymbol{Rev}(G^V)$) is a linear function and is set to be the sum of all virtual node capacity and virtual link bandwidth. Weight factors (α and β) are used to balance different types of network resources. Formula 6 is the per-unit revenue of a G^V.

$$\boldsymbol{Rev}(\mathrm{G^V})=\alpha\bullet\sum_{M\in \mathrm{N^V}}\mathbf{C}_M^{\mathrm{V}}+\beta\bullet\sum_{M\in \mathrm{L^V}}\mathbf{B}_{MN}^{\mathrm{V}} \tag{6}$$

Though Formula 5 and 6 give an insight into how much revenue can be earned by accepting a VN, it is useless without knowing the amount of consumed substrate resources. Therefore, Formula 7 gives the per-unit cost of embedding a VNR. Weight factors (γ and δ) are used to balance different types of network resources. $\boldsymbol{H}_{MN}^{mn}$ are the number of substrate links in path $\boldsymbol{mn}$, mapping the virtual link $\boldsymbol{MN}$. With respect to the total cost of embedding a VNR $\boldsymbol{G}^V$ in a continuous time event, it is the product of its per-unit cost and duration time, similar to the previous Formula 5.

$$\boldsymbol{Cos}(\boldsymbol{G}^V)=\gamma \underset{M\ \mathrm{N^V}}{} \mathbf{C}_M^{\mathrm{V}}+\eta \underset{MN\ \mathrm{L^V}\ \mathbf{mn}\ \mathrm{P^S_{mn}}}{} \boldsymbol{H}_{MN}^{mn}\ \mathbf{B}_{MN}^{\mathrm{V}} \tag{7}$$

The long-term average VNR acceptance ratio is another important metric in VNE research. In this paper, it is determined by the number of successfully accepted VNRs **a'** and the number of total proposed VNRs **a** in the long term. It is shown in Formula 8 below.

$$A^{VN}=\mathbf{a'}/\mathbf{a} \tag{8}$$

Other universal VNE metrics (e.g. average node / link utilization in a continuous time event), used in this paper, are not carefully introduced due to the limited page. Readers can refer to [8] and [9] for a detailed VNE metric definition, too.

IV. Proposed Algorithm *VNE-TAGRD*

The proposed *VNE-TAGRD* algorithm is detailed in this section. Three fundamental and essential topology attributes, adopted in *VNE-TAGRD*, are introduced and quantified firstly. Then the novel node-ranking approach is presented. Next, the greedy node mapping is implemented based on the novel node-ranking approach. After completing the node mapping, the shortest-path (SP) approach follows. The time complexity of *VNE-TAGRD* is also presented to prove that *VNE-TAGRD* algorithm can run in polynomial time and can be simulated in a continuous time event theoretically.

A. Selected Topology Attributes

In this subsection, three fundamental and essential topology attributes are introduced [28][29]. As known in reference [19], different topology attributes will have different critical effects on the embedding of each VNR [28]. Each topology attribute enables to measure the relative importance of each node from the corresponding respect. Derived from the previous studies (e.g. [19][20]), three fundamental and essential topology attributes [28][29], adopted in our novel node-ranking approach and *VNE-TAGRD*, are selected and introduced in this subsection. Other topology attributes, such as “eigenvector centrality” and “Katz centrality”, are used in directed networks [33] and are not adopted in our *VNE-TAGRD* algorithm.

1) **Degree** of a node $\boldsymbol{m}$: Formula 9 is the definition of the degree of node $\boldsymbol{m}$ in a given network. It is determined by the function **totlink( )**, counting the number of adjacent links of node $\boldsymbol{m}$ in the network.

$$\boldsymbol{Degree}(\boldsymbol{m})=\mathbf{totlink}(\boldsymbol{m}) \tag{9}$$

2) **Strength** of a node $\boldsymbol{m}$: Formula 10 presents the definition of strength of node $\boldsymbol{m}$ in a given network. It is determined by the function **totband( )**, counting the sum of all adjacent link bandwidth of node $\boldsymbol{m}$ in the network.

$$\boldsymbol{Strength}(\boldsymbol{m})=\mathbf{totband}(\boldsymbol{m}) \tag{10}$$

3) **Distance** between two nodes ($\boldsymbol{m}$ and $\boldsymbol{n}$): Formula 11 presents the distance between any two nodes ($\boldsymbol{m}$ and $\boldsymbol{n}$) in the given network. There must exist at least one straight path connecting the node $\boldsymbol{m}$ and node $\boldsymbol{n}$. It is determined by the function $\boldsymbol{ED}$**( )**, representing the ***Euclidean Distance*** (loop-free and shortest path) between nodes $\boldsymbol{m}$ and $\boldsymbol{n}$. Generally, nodes $\boldsymbol{m}$ and $\boldsymbol{n}$ are both defined by $\boldsymbol{x}$ and $\boldsymbol{y}$ coordinates, ($\boldsymbol{X_m}$, $\boldsymbol{Y_m}$) and ($\boldsymbol{X_n}$, $\boldsymbol{Y_n}$). The ***Euclidean Distance*** of nodes $\boldsymbol{m}$ and $\boldsymbol{n}$ is detailed in Formula 12.

$$\boldsymbol{Dis}(\boldsymbol{m},\boldsymbol{n})=\boldsymbol{ED}(\boldsymbol{m},\boldsymbol{n}) \tag{11}$$

$$\boldsymbol{ED}(\boldsymbol{m},\boldsymbol{n})=\sqrt{(\boldsymbol{X}_m-\boldsymbol{X}_n)^2+(\boldsymbol{Y}_m-\boldsymbol{Y}_n)^2} \tag{12}$$

B. Node-Ranking Approach

In this subsection, the novel node-ranking approach, adopted in the *VNE-TAGRD*, is detailed. The node-ranking approach, similar to the *PageRank* algorithm, is able to estimate the embedding ability of each substrate node for accommodating each given VNR. At first, the novel metric “**Node Value**” (***NoV***), quantifying three topology attributes and global network resources simultaneously, is defined. Global network resources considered in this paper are node capacity, node location, link bandwidth and link propagation delay.

Inspired by the ***Coulomb’s law*** [34] in electromagnetism area and the ***Newton’s law*** [35] in gravitational field, the interactions between any two discrete objects can be quantified (Formula 13). Therefore, the authors formulate the Formula 14 to quantify the interaction between two nodes, $\boldsymbol{m}$ and $\boldsymbol{n}$, in the given network. Note: there must exist at least one loop-free path from $\boldsymbol{m}$ to $\boldsymbol{n}$. If so, Formula 14 can be adopted to qualify the interaction between node $\boldsymbol{m}$ and node $\boldsymbol{n}$ in the node-ranking stage.

$$\left|\boldsymbol{F}_{1,2}\right|=\boldsymbol{k}\bullet\frac{\left|\boldsymbol{q}_1\bullet\boldsymbol{q}_2\right|}{\boldsymbol{Dis}(\boldsymbol{q}_1,\boldsymbol{q}_2)^2} \tag{13}$$

where k is a constant, q_1 and q_2 are the weights or electronic charges of two objects, $Dis(q_1, q_2)$ represents the ***Euclidean Distance*** between objects q_1 and q_2 [34].

$$NoV_{m,n} = \alpha \bullet \frac{RB_m \bullet RB_n}{Dis(m,n)^2 \bullet D_{mn}^2} \quad (14)$$

where α is also a constant. RB_m is the resource block of node m (Formula 15). Formula 15 below defines ***the resource block (RB)*** for node m and aims to strength the resource measurement of node m in the network. C_m is the node capacity of node m. **Degree** and **Strength** topology attributes are also adopted to define RB. It is same to ***the resource block*** RB_n of node n. $Dis(m, n)$ in Formula 14 and 15 represents the ***Euclidean Distance*** between node m and n **(**Note: at least one straight path connecting node m and node n; no loop**)**. D_{mn} represents the propagation delay of substrate path mn. For simplicity, the propagation delay of each substrate link is set to be one time unit in our paper. That is to say, the propagation delay of one selected path mn is equal to the number of substrate links in the substrate path mn. To achieve the node m's NoVs with remaining nodes in the whole network, the authors further propose the Formula 16. Formula 17 aims to normalize *the **NoV** percentage* of node m in the whole network. Specifically, the ***NoV** percentage* (*NoV %*) of node m increases with the node m's available capacity, node degree and node strength increasing.

$$RB(m) = C_m \quad Degree(m) \quad Strength(m) \quad (15)$$

$$NoV_m = \sum_{m \quad n,n \quad G} NoV_{m,n} \quad (16)$$

$$NoV\%(m) = \frac{NoV(m)}{\sqrt{\sum_{m \in G} NoV(m)^2}} \quad (17)$$

With calculating all percentage values of ***NoV%*** **(m)** in the given network, all percentage values of the network make up an initial node-ranking vector T_0. The vector T_0 is presented as follows:

$$T_0 = \left(NoV\%(1),\ NoV\%(2),\ \dots,\ NoV\%(|N|),\right)^T$$

On the basis of NoV calculation and the initial node-ranking vector T_0, the eventual node-ranking values of the given network G (either a substrate network or a VNR) will be in a stable state and calculated in a recursive manner. For each node m in the given network, its eventual node-ranking value is set as r_m and is presented in Formula 18.

$$r_m = (1-d) \bullet RB\%(m) + d \bullet \sum_{m \quad n,n \quad N(m)} NoV_{m,n} \bullet r_n \quad (18)$$

where d is the damping factor within (0,1). *N(m)* indicates the set of all nodes having a loop-free path with the node m in the given network G. The ***RB%*****(m)** is the normalized resource block of node m in the network. Its detailed version is similar to what are presented in the Formula 17. As can be drawn from Formula 18, the node-ranking value of any node m increases with the node's available resource block and the node values of its nearby nodes increasing. Higher ranked substrate nodes contribute to larger successful probability in the VNR embedding stage.

To represent all nodes in a vector R, the traffic form of all node-ranking values is developed and shown in Formula 19 below.

$$R = (1-d) \bullet RB\% + d \bullet M \bullet R \quad (19)$$

where the $R=(r_1, r_2, \dots r_m \dots, r_{|N|})^T$. The representation of $RB\%$ is $(RB\%_{(1)}, RB\%_{(2)}, \dots RB\%_{(m)} \dots, RB\%_{(|N|)})^T$. M is the transition matrix of dimension |N|•|N|. d is the damping factor within (0,1), as mentioned above. Each element in the matrix M is detailed in Formulas 17 and 18. If there exists no path directly connecting node m and node n, the corresponding element in the matrix M is set to be 0.

By referring to the matrix theory [36], the unique solution of Formula 19 can be directly given by follows:

$$R = (1-d) \bullet (I - d \bullet M)^{-1} RB\% \quad (20)$$

For convincing researchers that Formula 20 is the final unique solution of Formula 19 in the given network G, we are to provide the proof below.

Proposition: Matrix (I-d•M) is reversible and a final unique solution (Formula 20) can be obtained from Formula 19.

Proof: With Formula 16 and 17, it is easy to know that the sum of $NoV_{m,n}$ is equal to 1 (Formula 21).

$$\sum_{m=1}^{|G|} NoV_{m,n} = 1 \quad (21)$$

Therefore, $\|M\| \le 1$ can be easily concluded (Gershgorin Circle Theorem [37]). It is difficult to prove that matrix (I-d•M) is reversible directly.

Let us make an assumption that matrix (I-d•M) is singular. It is easy to know that the linear system equations (I-d•M)•**r**=0 have non-zero solutions. Let $\mathbf{r_0}$ be one selected non-zero solution of the linear system equations. Then it is easily to get that d•M•$\mathbf{r_0}$=$\mathbf{r_0}$.

Therefore, we can easily get the result of $\|\mathbf{r_0}\| = \|d \bullet M \bullet \mathbf{r_0}\| \le d \bullet \|M\| \bullet \|\mathbf{r_0}\|$. Thus getting $\|M\| \ge (1/d) > 1$ in the end. However, the conclusion of $\|M\| > 1$ violates the conclusion of Formula 21 ($\|M\| \le 1$). Therefore, we can make the eventual conclusion that matrix (I-d•M) is reversible.

With known that matrix (I-d•M) is reversible, it is easily to calculate the unique solution of Formula 19. The unique solution is given below.

$$R = (1-d) \bullet (I - d \bullet M)^{-1} RB\% \quad (22)$$

Two important points must be pointed out in proving the eventual solution (Formula 20). One point is to prove the uniqueness of the final node-ranking solution R. The other point is to prove that matrix (I-d•M) is reversible. The proof of that matrix (***I*-*d*•*M***) is reversible is extremely important before calculating the final node-ranking solution ***R***.

By referring to reference [37], the complexity of directly calculating out Formula 20 is $O(|N|^3)$. The time complexity will be very complex with the scale of network scenario expanding. As the simulation of this paper is conducted in a continuous time event, backtracking and recursion methods can not be applied to calculate Formula 20. Therefore, an iterative approach can be adopted. Through *k* iterations, it is easy to converge to a stable solution and get a final solution of Formula 20, same to the Jacobi algorithm for solving the linear system equations [36]. Therefore, corresponding complexity of the iteration-based node-ranking approach is $O((|N|^2)*\log(1/\delta))$. δ is a small positive number to ensure the number of iterations. The procedures of node-ranking approach are detailed in ***Algorithm 1*** below.

Algorithm 1 **Novel Node-Ranking Approach**

Input: Network G=(N, L), a small positive number δ
Output: Node-Ranking Vector ***R*** of the given Network ***G***

1. Calculate matrix ***M*** and initial vector $\boldsymbol{R_0}$ ($\boldsymbol{T_0}$)
2. Define the iteration number *k*, *k*=0.
3. Define the variable w, w=∞.
4. ***while*** w ≥ δ ***do***
5. $\boldsymbol{R_{k+1}}$=(1-d)****RB%***+d****M****$\boldsymbol{R_k}$;
6. w= ||$\boldsymbol{R_{k+1}}$ **-** $\boldsymbol{R_k}$||;
7. *k*=*k*+1;
8. ***end while***
9. $\boldsymbol{R=R_{k+1}}$

C. Greedy Node Mapping

In the *VNE-TAGRD* algorithm, the node mapping of a given VNR works in a greedy way. The status of the whole substrate network is backup before embedding one proposed VNR. Then all nodes of both the substrate network and the proposed VNR are sorted in the decreasing order, according to the node-ranking values calculated by the novel node-ranking approach in above subsection. As the node-ranking value indicates the embedding ability of the corresponding node, virtual nodes are embedded onto the substrate nodes by processing the two sorted node lists (the substrate network and the VNR) with a strategy similar to the well-known *merge-sort* algorithm [37]. Therefore, the virtual node with the highest node-ranking value among the remaining virtual ones of the VNR will always be embed onto the substrate node that also has the highest node-ranking value among the remaining substrate ones, whose available resource meets the VNR demands (node location and node capacity are considered in node mapping stage). For instance, if the node capacity demand cannot be satisfied by any of the remaining substrate nodes, the VNR will be marked as rejected. If all virtual nodes of the proposed VNR are embedded successfully, the node capacity of the corresponding substrate nodes will be further updated. The greedy node mapping procedure are presented in the ***Algorithm 2***. The complexity of the greedy node mapping is $O(|N^S||N^V|)$ [37].

D. Shortest Path Link Mapping

With embedding all virtual nodes of the proposed VNR, the virtual links of the VNR demands to be embedded. For the link mapping stage of the proposed VNR, the universal shortest-path algorithm is applied in *VNE-TAGRD* so as to minimize the total consumed substrate link bandwidth. Particularly, the virtual links of each proposed VNR are processed one by one. For each virtual link of the given VNR, the *Dijkstra's algorithm* [37] is adopted to find the shortest path between the two corresponding substrate nodes in the substrate network. In addition, to further improve the efficiency of the *VNE-TAGRD*, a pruning procedure, deleting all the substrate links in the substrate network that do not have enough bandwidth for the corresponding virtual link, is conducted in the link mapping stage. In addition, to each virtual link of the VNR, the propagation delay of its selected path must be within the required virtual link propagation delay at the same time. If the link mapping of the given VNR fails (i.e. not all virtual links of the same VNR are embedded successfully), the status of the who substrate network is restored. The VNR is therefore rejected. The ***Algorithm 2*** also details the shortest-path link mapping algorithm. The complexity of the shortest-path link mapping algorithm is $O(|L^S||L^V|\log|N^S|)$ in our paper.

At the end of this section, it is of great importance to calculate the total time complexity of *VNE-TAGRD* algorithm, when embedding one VNR. The total time complexity of *VNE-TAGRD* can be calculated by adding up the time complexities of the three main steps (novel node-ranking approach, greedy node mapping and SP link mapping). Therefore, the total time complexity is $O((|N^S|^2+|N^V|^2)*\log(1/\delta)+|N^S||N^V|+|L^S||L^V|\log|N^S|)$. The *VNE-TAGRD* can be solved in polynomial time and can be simulated in a continuous time event (Section V).

Algorithm 2 **Algorithm** *VNE-TAGRD*

Input: Arrived VNs in a 1000 time units and the SN
Output: Results of VNRs' embedding and unmapped VNRs

1. ***while*** there is unprocessed VNRs ***do***
2. Take VNR_i that has the smallest revenue [16]
4. *Algorithm 1*
5. ***for*** each virtual node of the VNR_i ***do***
6. Select the virtual node with the highest node-ranking value of the VNR_i and map it to the substrate node with the highest node-ranking value of the SN, with meeting the constraints of node location and node capacity. To the given VNR_i, two virtual nodes can not share the same substrate node
7. ***end for***
8. ***if*** the virtual node mapping of VNR_i *succeeds* ***then***
9. The pruning procedure is conducted for deleting all the substrate links in the substrate network that do not have enough bandwidth for the corresponding virtual link. Select the shortest path to map all virtual links, with fulfilling all link bandwidth demands of the VN_i. The propagation delay of each selected path must be within the required link propagation delay of its mapped virtual link at the same time
10. ***else*** reject the VNR_i and throw the VNR_i into the waiting queue
11. ***end if***
12. ***end while***

V. SIMULATION EVALUATION

This section presents the simulation parameter settings followed by the simulation results. This section elaborates on quantifying the efficiency of *VNE-TAGRD* algorithm. Five typical and state-of-the-art heuristic algorithms are selected to make up the simulation altogether.

A. Simulation Settings and Parameters

To evaluate the *VNE-TAGRD*, a continuous event simulator in JAVA has been implemented. Since evaluating multiple different VNE algorithms in a same platform simultaneously is still an emerging field, the authors of this paper conduct the simulation work in a self-developed platform. The self-developed platform is called as the '*Simulation Platform for Scotfield Cao*' [38]. Some codes are available to public free of charge [39].

In this paper, the shared substrate network is generated using *the Waxman method* [40], integrated as a module in '*Simulation Platform for Scotfield Cao*'. The substrate network is considered medium-scaled in VNE research. The number of substrate nodes is set to be 60. Set α=0.4 and β=0.3 in the substrate network. The node capacity and link bandwidth of substrate nodes and links are integers uniformly distributed between 50 and 100. Each substrate node is randomly located within a uniformly distributed position between 0 and 100 on x and y coordinates. Each substrate link propagation delay is set to be 1 in this paper, as talked in Section IV.

VNRs are also generated by *the Waxman method*. VNs arrive in a *Poisson process* and are evaluated by setting VNs arrival rate 5 per 1000 time units. Each VN has an exponentially distributed lifetime with an average value of μ=1000 time units. These parameters are universal and typical in VNE algorithm research area.

To each VNR, the number of nodes is an integer and uniformly distributed between 2 and 10. Link connectivity parameters of each VN are same to what are set in substrate network. Node capacity and bandwidth requirements of virtual nodes and links are integers uniformly distributed between 1 and 20. Virtual nodes are randomly located within a uniformly distributed position between 0 and 100 on x and y coordinates. Allowed maximum deviation of each virtual node is an integer and uniformly distributed between 3 and 8. Each virtual link propagation delay requirement is also an integer and uniformly distributed between 1 and 4.

Simulations are run for about 100000 time units. That's to say, 500 VNRs on average in total are to be embedded. α, β, γ and δ in Formula 6 and Formula 7 are all set to be 1. δ in node-ranking approach (*Algorithm 1*) is set to be 0.00001 [24]. The value of **d** in this paper is set to be 0.85, same to the value setting in *PageRank* algorithm [25]. All simulations for different VNE algorithms are run on a Window 8 Desktop, with an Intel® Core (TM) CPU i7-4790@3.6GHz Processor and 16.00G RAM Machine.

B. Compared Algorithms

Six heuristic VNE algorithms make up the simulation part totally. Besides of the proposed *VNE-TAGRD*, remaining algorithms are enumerated in Table II, along with short descriptions. These algorithms are typical, latest and related to our algorithm and are all slightly modified to fit into the simulation of our paper (e.g. node location demands and link propagation delay considered).

Table II
COMPARED ALGORITHMS

Notation	Description
G-SP [16]	Greedy Node Mapping and Shortest Path Link Mapping
RW-SP [17]	Random Walk Node Mapping and Shortest Path Link Mapping
VNE-DCC [20]	Virtual Network Embedding with Degree and Clustering Coefficient Information
VNE-RD [21]	Virtual Network Embedding by Revenue-Driven
VNE-NRM [48]	Virtual Network Embedding by Node Ranking Metric
VNE-TAGRD	Virtual Network Embedding by Topology Attributes and Global Network Resources Driven

C. Simulation Results

In this subsection, the simulation results are plotted to highlight the *VNE-TAGRD*. Fig. 3 presents the average VNR acceptance ratio as a function of time while Fig. 4 shows the average revenue to cost ratio as a function of time. Concerning illustration is also presented below. Both two figures aim to directly prove the efficiency and effectiveness of *VNE-TAGRD* in the long term. Other important metrics, such as average node utilization and link utilization, are selected and plotted in Fig. 5 and Fig. 6 to highlight the advantage of *VNE-TAGRD* algorithm indirectly.

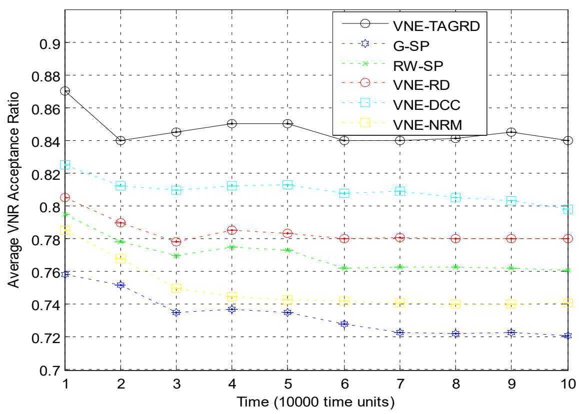

Fig. 3 Average VNR Acceptance Ratio

1) Average VNR Acceptance Ratio: Fig. 3 is the average VNR acceptance ratio as a function of continuous time. The VNR acceptance ratio is an important metric to evaluate different VNE algorithms' mapping abilities in a continuous time event. Observed from the Fig. 3, the average VNR acceptance ratio of all algorithms almost decays with the variation on time. This decay shows that there are no infinite substrate resources for embedding more and more given VNRs. In addition, *VNE-TAGRD* outperforms all selected heuristics. The difference between the best behaved heuristic and *VNE-TAGRD* is approaching 5%. It runs as expected because the *VNE-TAGRD* takes several essential topology attributes and global network resources into account simultaneously. The relationships with other nodes is further explored by *VNE-TAGRD*. An efficient node mapping is therefore likely to be achieved. To the remaining heuristic algorithms (*G-SP, RW-SP, VNE-RD, VNE-DCC and VNE-NRM*), only local resources and single topology attribute are considered in the node mapping stage. The node embedding of each VNR is not efficient in many cases. Therefore, a feasible mapping is tried to be found. Comparing with the heuristics, the *VNE-TAGRD* behaves best generally.

2) Average Revenue to Cost Ratio: Fig. 4 illustrates the average revenue to cost ratio as a function of time. Derived from the Fig. 4, the average revenue to cost ratio of all selected algorithms decreases with the simulation time increasing. To *G-SP*, the reason for lowest revenue to cost ratio contributes to the fact that adjacent virtual nodes are mapped onto substrate node, far from each other. Thus leading to large amount of unnecessary substrate resource consumption. To the other heuristic algorithms (*RW-SP, VNE-RD*, *VNE-DCC* and *VNE-NRM*), only considering local resources and relationships with nearby nodes, extra required resources are saved. Comparing with the *VNE-TAGRD* algorithm, *VNE-DCC* and *VNE-RD* are not able to behave better. It is owing to that multiple essential topology characteristics and global network resources are considered in *VNE-TAGRD* simultaneously. The relationship of each node with all other nodes are fully explored, too. The novel node-ranking approach of *VNE-TAGRD* further ensures the efficiency of substrate network resources usage in the long run.

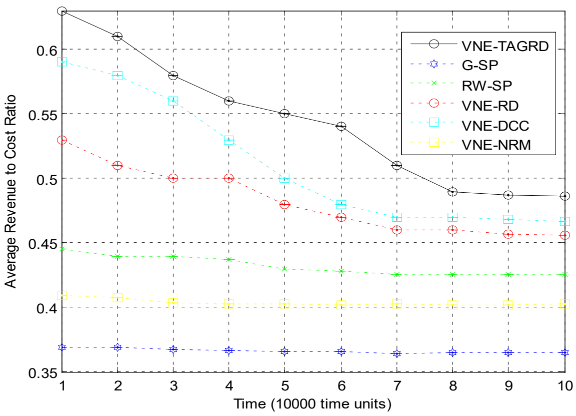

Fig. 4 Average Revenue to Cost Ratio

3) Average Node and Link Utilization: Average node and link utilization as a function of time are depicted in Fig. 5 and Fig. 6. With the number of VNRs increasing, the node and link utilization of all selected algorithms increase, too. To the node utilization, depicted in Fig. 5, the *VNE-TAGRD* has an apparent advantage over the remaining heuristic algorithms. The reason for *VNE-TAGRD* having a larger node utilization than the heuristics lies to *VNE-TAGRD*'s ability of accepting more VNRs than the other heuristics, considering more topology characteristics and global network resources in the node mapping stage. When the number of VNRs is incresing, the algorithm *VNE-TAGRD*, is able to embed VNRs more effectively and loads the substrate nodes to their full capacity. However, the link utilization in Fig. 6 does not have the same behavior for all algorithms, as shown in Fig. 5 for the node utilization. All six algorithms behave similar to each other. Shortest path approach is adopted in the link mapping stage, to all these six VNE algorithms. Therefore, all six algorithms run similarly.

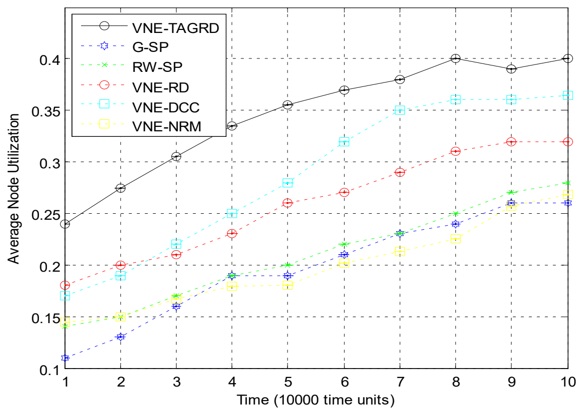

Fig. 5 Average Node Utilization

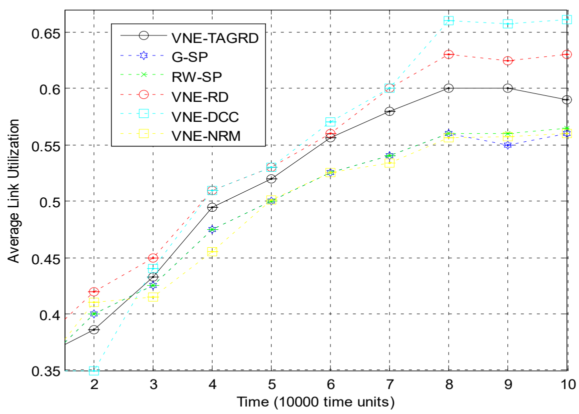

Fig. 6 Average Link Utilization

VI. Conclusion and Future Work

This paper proposes an efficient heuristic algorithm *VNE-TAGRD* to embed virtual networks in a continuous time event. The *VNE-TAGRD* adopts a novel node-ranking approach, on the basis of three fundamental topology attributes and global network resources, to rank all substrate and virtual nodes before conducting each VNR embedding. The novel node-ranking approach, stimulating from the well-known *Google PageRank* algorithm, is used to assist the greedy node mapping. Shortest-path approach is adopted in the following link mapping stage. When running *VNE-TAGRD* to embedding a VNR, four different node-link constraints must be fulfilled, the virtual link propagation delay constraint included. The virtual link propagation delay is considered as a node-link constraint for the first time in VNE research area. Simulation results reveal that *VNE-TAGRD* algorithm outperforms five typical and state-of-the-art heuristic algorithms, in terms of the long-term average VNR acceptance ratio and average revenue to cost ratio.

For the future work, there are still a number of issues that remain to be done. First of all, it is to upload the *VNE-TAGRD* in a real testbed environment and evaluate the *VNE-TAGRD* through a prototype implementation. The coding and simulation of our paper are all conducted on our self-developed VNE platform "*Simulation Platform for Scotfield Cao*" [39]. The virtual link propagation delay performance of each given VNR is to be further explored and analyzed, on the basis of our programming method *CAN-A* [47].

ACKNOWLEDGMENT

This work will be presented in part at the IEEE International Conference on Wireless Communications and Signal Processing, October 11–13, 2017, Nanjing, China. This paper is partly supported by the National Basic Research Program of China (973 Program) under Grant 2013CB329104, the National Natural Science Foundation of China under Grant 61372124 and 61427801. The work of G. Zheng was supported by the UK EPSRC under grant number EP/N007840/1.

REFERENCES

[1] J. S. Turner and D.E. Taylor, "Diversifying the internet," in *Proc. IEEE GLOBECOM'05*, St. Louis, MO, USA, 2005.

[2] N. M. M. K. Chowdhury, "Network virtualization: State of the art and research challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20-26, Jul. 2009.

[3] L. Peterson, S. Shenker, J. Turner *et al.*, "Overcoming the internet impasse through virtualization," *IEEE Computer*, vol. 38, no. 4, pp. 34-41, Apr. 2015.

[4] Cisco. Network virtualization: The new building blocks of network design Available:http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/

[5] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service(iaas)," *Int J. Eng. And Information Technology*, vol. 2, pp. 60-63, 2010.

[6] N. Chowdhury and R. Boutaba, "A Survey of network virtualization," *Computer Networks (Elsevier)*, vol. 54, no. 5, pp. 862-876, 2010.

[7] D. Andersen, "Theoretical approaches to node assignment," 2002 [Online].

[8] A. Fischer, J. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888-1906, 4th Quart. 2013.

[9] H. Cao, L. Yang *et al.*, "Exact Solutions of VNE: A Survey" *China Commun.*, vol. 13, no. 6, pp. 48-62, June 2016.

[10] J. Botero *et al.*, "Energy efficient virtual network embedding," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 756-759, May 2012.

[11] M. Melo *et al.*, "Optimal Virtual Network Embedding: Node-Link Formulation," *IEEE Trans. Netw. and Serv. Manag.*, vol. 10, no. 4, pp. 356-368, Dec. 2013.

[12] I. Houidi, W. Louati, and D. Zeghlache, "A distributed virtual network mapping algorithm," in *Proc. 2008 IEEE ICC*, pp. 5634-5640.

[13] M. Chowhury, M. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206-219, Feb. 2012.

[14] A. Jarray and A. Karmouch, "Decomposition approaches for virtual network embedding with one-shot node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 1012-1025, Jun.2015.

[15] R. Mijumbi et al., "A Path Generation Approach to Embedding of Virtual Networks," *IEEE Trans. Netw. and Serv. Manag.*, vol. 12, no. 3, pp. 334-348, Sep. 2015.

[16] M. Yu, Y. Yi, *et al.*, "Rethinking virtual network embedding: substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17-29, Mar. 2008.

[17] X. Cheng *et al.*, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38-47, Apr. 2011.

[18] S. Su, *et al.*, "Energy-Aware Virtual Network Embedding," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1607-1620, Oct. 2014.

[19] M. Feng, J. Liao, J. Wang, S. Qing, and Q. Qi, "Topology-aware virtual network embedding based on multiple characteristics," in *2014 IEEE Conference on Communications (ICC)*, pp. 2956-2962.

[20] P. Zhang, H. Yao and Y. Liu, *et al.*, "Virtual network embedding based on the degree and clustering coefficient information," *IEEE Access,* vol. 4, pp. 8572-8580, 2016.

[21] L. Gong, Y. Wen , Z. Zhu and T. Lee, "Revenue-Driven Virtual Network Embedding Based on Global Resource Information," in *Global Communications Conference (GLOBECOM), 2013 IEEE,* Dec. 2013, pp. 2294-2299.

[22] Long Gong, Huihui Jiang, Yixiang Wang and Zuqing Zhu, "Novel Location-Constrained Virtual Network Embedding(LC-VNE) Algorithms Towards Integrated Node and Link Mapping," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3648-3661, 2016.

[23] M. Brinkmeier, "PageRank revisited," *ACM Trans. Internet Technol.*, vol. 6, no. 3, pp. 282-301, Aug. 2006.

[24] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proc. of WWW 1998*, pp. 107-117, 1998.

[25] F. Chung, "A brief survey of PageRank Algorithms," *IEEE Trans. Netw. Sci. and Engin.*, vol. 1, no. 1, pp. 38-42, 2014.

[26] M. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, "Multi-path link embedding for survivability in virtual networks," *IEEE Trans. Netw. and Serv. Manag.*, vol. 13, no. 2, pp. 253-266, 2016.

[27] [Online]. Available: http://aws.amazon.com/ec2/#pricing

[28] M. Newman, Networks: An Introduction. *Oxford University Press*, Oxford, UK, 2010.

[29] L. Freeman, The development of social network analysis. Empirical Press Vancouver, BC, 2004.

[30] A. F. Molisch, *Wireless Local Area Networks*, New York, NY, USA: Wiley-IEEE , 2012.

[31] J. Stribling, V. Arunarthi, C. Knittle, D. Murayama and M. Emmendorfer, "Implementing Qos in SIEPON," *IEEE Commun. Mag.*, vol. 50, no. 9, pp. 128-135, 2012.

[32] S. Andreev, O. Galinina, A. Pyattaev *et al.*, "Exploring Synergy between Communications, Caching, and Computing in 5G-grade Deployments," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 60-69, 2016.

[33] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: generalizing degree and shortest paths," *Social Networks*, vol.32, no.3, pp.245-251. 2010.

[34] W. Sousa, R. Oliveira, ", Coulomb's law discretization method: a mew methodology of spatial discretization for the radial point interpolation method," *IEEE Ant. And Pro. Mag.*, vol. 57, no. 2, pp. 277-293, Apr. 2015.

[35] R. Bonsal, "AP-S turnstile- invoking heaviside in dense of Newton's third law,," *IEEE Ant.and Pro. Mag.*, vol. 48, no. 4, pp. 118-119, Oct. 2006.

[36] G. Golub and C. van Van Loan, Matrix Computations *(Johns Hopkins studies in Mathematical Sciences) (3rd Edition)*, Johns Hopkins University Press, Oct. 1996.

[37] T. H. Cormen, C. Stein, R. Rivest and C. Leiserson, *Introduction to Algorithms, 2nd ed.* McGraw-Hill Higher Education, 2001.

[38] H. Cao, S. Hu and L. Yang, "New Functions added to ALEVIN for evaluating virtual network embedding," in *2nd Internation Conference on Computer and Communications, 2016 IEEE, Oct*. 2016, pp. 2411-2414.

[39] http://pan.baidu.com/s/1gekPZrl.

[40] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol 6, no. 9, pp. 1617-1622, 1988.

[41] M. Rahman and R. Boutaba, "SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization,"*IEEE Trans. Netw. and Serv. Manag.*, vol. 10, no. 2, pp. 105-118, Jun. 2013.

[42] S. Chowdhury *et al.*, "Dedicated Protection for Survivable Virtual Network Embedding,"*IEEE Trans. Netw. and Serv. Manag.*, vol. 13, no. 4, pp. 913-926, May. 2016.

[43] D. Dietrich, A.. Rizk, and P. Papadimitriou, "Multi-provider virtual network embedding with limited information disclosure,"*IEEE Trans. Netw. and Serv. Manag.*, vol. 12, no. 2, pp. 188-201, May. 2015.

[44] "GLPK(GNU Linear Programming Kit)," Jul. 2017. Available: http://www.gnu.org/software/glpk/.

[45] "IBM ILOG Optimization Products," Jul. 2016. Available: www=01.ibm.com/software/websphere/products/optimization.

[46] A. Schrijiver, *Theory of Linear and Integer Programming*. New York, NY, USA: Wiley, 1998.

[47] H. Cao, Y. Zhu, G. Zheng and L. Yang, "*A Novel Optimal Mapping Algorithm with Less Computational Complexity and Average Virtual Link Delay for Virtual Network Embedding*," submitted to *IEEE Trans. Netw. and Serv. Manag.*

[48] P. Zhang, H. Yao and Y. Liu, "Virtual network embedding based on computing, network and storage resource constraints," *IEEE IoT Journal,*vol. pp, no. 99, pp. 1-1, 2017.