Virtual Network Embedding with Multi-attribute Node Ranking Based on TOPSIS

Shuiqing Gong, Jing Chen, Siyi Zhao and Qingchao Zhu
College of Information and Navigation, Air Force Engineering University
Xi'an, Shanxi 710077 - China
[e-mail: gsq0121@126.com]
*Corresponding author: Jing Chen

Received September 2, 2015; revised December 5, 2015; accepted December 23, 2015; published February 29, 2016

Abstract

Network virtualization provides an effective way to overcome the Internet ossification problem. As one of the main challenges in network virtualization, virtual network embedding refers to mapping multiple virtual networks onto a shared substrate network. However, existing heuristic embedding algorithms evaluate the embedding potential of the nodes simply by the product of different resource attributes, which would result in an unbalanced embedding. Furthermore, ignoring the hops of substrate paths that the virtual links would be mapped onto may restrict the ability of the substrate network to accept additional virtual network requests, and lead to low utilization rate of resource. In this paper, we introduce and extend five node attributes that quantify the embedding potential of the nodes from both the local and global views, and adopt the technique for order preference by similarity ideal solution (TOPSIS) to rank the nodes, aiming at balancing different node attributes to increase the utilization rate of resource. Moreover, we propose a novel two-stage virtual network embedding algorithm, which maps the virtual nodes onto the substrate nodes according to the node ranks, and adopts a shortest path-based algorithm to map the virtual links. Simulation results show that the new algorithm significantly increases the long-term average revenue, the long-term revenue to cost ratio and the acceptance ratio.

Keywords: Network virtualization, virtual network embedding, node attribute, rank, TOPSIS

This work was jointly supported by the National Natural Science Foundation of China (No. 61201209, 61401499), the Natural Science Foundation of Shanxi (No. 2013JQ8013, 2015JM6340).

http://dx.doi.org/10.3837/tiis.2016.02.005 **ISSN : 1976-7277**

1. Introduction

The Internet has been playing a vital role in modern society over the years. However, due to the existence of multiple Internet service providers (ISPs) with conflicting purposes and policies, new network services and innovative technologies are hard to deploy in current Internet, and even the essential modifications to the architecture of the Internet meet resistance, which are known as the Internet ossification problem [1]. To overcome this impasse, network virtualization has been put forward as a fundamental ingredient of the future Internet paradigm, which allows multiple heterogeneous virtual networks (VNs) to coexist on a shared substrate network (SN) [2-4]. In a network virtualization environment, traditional ISPs are decoupled into two independent entities: infrastructure providers (InPs) and service providers (SPs). InPs are responsible for managing the physical SNs, while SPs are responsible for creating VNs by leasing resources from InPs and offering customized end-to-end services to users. Such an environment will facilitate the innovation of new network architectures free of the inherent limitations of the current Internet.

Since SNs have limited network resources, in order to increase the revenue of InPs, it is crucial to efficiently assign substrate resources to virtual network request (VNR) while satisfying the constraints of virtual nodes and links, which is known as the virtual network embedding problem (VNE) [5]. As the main challenge in network virtualization, the VNE problem is NP-hard [6] and numerous algorithms have been proposed to solve it [7-20]. Generally, The VNE consists of two mapping stages: the node mapping where virtual nodes are mapped onto substrate nodes in a one-to-one manner while satisfying the constraints of nodes' resources, and the link mapping where virtual links are mapped onto loop-free substrate paths while satisfying the constraints of links' resources. Moreover, the former stage is more important as it is the basis of the latter one.

Most existing algorithms adopt greedy strategy to map the virtual nodes requiring more resources onto the nodes in the substrate network with more available resources in the node mapping stage [9-16]. The node ranking method is the key problem in the greedy node mapping. It evaluates the embedding potential of each node in the VNs and the SN, and ranks them in descending order. Then, virtual nodes are mapped onto substrate nodes according to their ranks. There are many factors that would affect the evaluation. However, previous algorithms evaluate the embedding potential of nodes simply by the product of their CPU resource, the total bandwidth resource of their adjacent links and their topological attributes in the network [13-16], which would result in an imbalance of these factors, and eventually lead to decreased utilization rate of resource of the SN. Moreover, the hops of substrate paths that the virtual links mapped onto should also be considered in the node mapping stage. Otherwise, two adjacent virtual nodes connected directly by a virtual link in the VN may be mapped onto the substrate nodes that far away from each other in the SN, resulting in the waste of bandwidth resources and restricting the ability of the SN to accept additional VNRs.

In this paper, we propose a novel VNE algorithm (TOP-VNE) with multi-attribute node ranking, which can efficiently solve the above problems. Five node attributes are introduced and redefined to measure the embedding potential of nodes for the VNE problem: "resource capacity", "communication capacity", "degree", "closeness", "correlation quality", where resource capacity and communication capacity reflect the local resources of nodes, degree and closeness reflect the local and global topological attributes of nodes in the network, respectively, and correlation quality considers the hops of substrate paths. Then we regard each node in the VN or SN as a solution, and each node attribute as a solution's evaluation criterion of importance, where the importance refers to the relative embedding potential of a node. In such a way, the evaluation of node importance is transformed into a multi-attribute decision making problem, and we adopt TOPSIS to solve it, resulting in a balanced importance evaluation among different attributes. The more important a node is in the network, the higher embedding potential it has. Based on the node ranks according to their importance, TOP-VNE gives a priority to mapping virtual nodes with high ranks onto substrate nodes with high ranks, and uses a shortest-path based algorithm to map virtual links. TOP-VNE combines the local network resource and multiple topological attributes together, and considers the balance among them, which enables better coordination between the two mapping stages and achieves larger utilization rate of resource.

In summary, the main contributions of this paper can be summarized as follows: (1) We introduce and extend five attributes of nodes, which take the local resources, different topological attributes and the hops of substrate paths into consideration, to measure their embedding potential for the VNE. (2) Base on the aforementioned five node attributes, we devise a multi-attribute node raking algorithm (TOP-MANR), which adopts TOPSIS to evaluate the importance of nodes and ranks them accordingly. TOP-MANR is leveraged in the VNE to solve the problem of the imbalance of different node attributes when ranking nodes in the VNs and the SN. (3) We propose a novel heuristic VNE algorithm (TOP-VNE), which consists of the TOP-MANR-based node mapping stage and the shortest path-based link mapping stage, (4) and extensive simulations are conducted to demonstrate the effectiveness and efficiency of our proposed VNE algorithm.

The rest of this paper is organized as follows. In section 2, we discuss the related work. Section 3 presents the network model and defines the VNE problem. Section 4 describes the multi-attribute node ranking algorithm. Our novel VNE algorithm is proposed in section 5. In section 6, we evaluate the proposed algorithm through extensive simulations. Section 7 concludes the paper.

2. Related Work

The VNE problem has been shown to be NP-hard due to the multiple constraints of nodes and links. Therefore, optimal results can only be achieved for small problem instances. Previous work on the VNE problem mainly relied on heuristic algorithms to obtain relative optimal solutions. Generally, the existing VNE algorithms can be divided into two categories: the one-stage embedding algorithm that the node mapping and the link mapping are completed at the same time, and the two-stage embedding algorithm that the node mapping is performed first and followed by the link mapping.

Cheng et al. [11] proposed a one-stage embedding algorithm through topology-aware node ranking. This algorithm considers the resources and topological attributes of nodes together, and applies the Markov Random Walk (RW) model to rank them. Based on breadth-first search, virtual nodes and links are mapped simultaneously according to the node ranks. Lu et al. [17] presented a novel VNE algorithm based on integer programming. They first built an augmented substrate graph by connecting each virtual node to its candidate substrate nodes, and then formulated the VNE problem as an integer program. The VNE problem is solved in one stage by GLPK and can obtain optimal results. However, this algorithm is not efficient for VNRs of complex topologies.

Most previous research focused on the two-stage embedding algorithms. Some of them mapped the virtual networks in two independent stages [9, 10]: the greedy node mapping which maps the virtual nodes with high resource demands onto the substrate nodes with large residual resources, and the shortest path-based link mapping which maps the virtual links onto the shortest substrate paths to reduce the cost. However, preselecting node mapping without taking its relation to the link mapping into account will restrict the solution space and lead to poor performance.

Therefore, recent works proposed a new type of two-stage VNE algorithm, which coordinates two mapping stages through considering the link mapping constraints in the node mapping stage. Chowdhury et al. [18] formulated the VNE problem as a mixed integer program through substrate network augmentation, and devised two online VNE algorithms by using deterministic and randomized rounding techniques, which achieve a better coordination between the two mapping stages. Wang et al. [13] introduced network centrality into the VNE problem and proposed two novel VNE algorithms based on closeness centrality, which enhances the collaboration of the two mapping stages. Cui et al. [14] considered the adjacent degree of virtual nodes and proposed a new algorithm based on maximum convergence-degree, which ensures the virtual nodes and links gather together after the embedding. Gong et al. [19] took the resources of the entire network into consideration, and formulated a novel metric called GRC to measure the embedding potential of substrate nodes. Based on this metric, they proposed a coordinated VNE algorithm, which maps virtual nodes in a greedy load-balance manner and adopts shortest path-based algorithm to map virtual links. Ding et al. [15] introduced the betweenness centrality into the VNE problem. They proposed an embedding algorithm based on real-time topological attributes, which maps virtual nodes according to the node ranking results and uses *k*-shortest path algorithm to map virtual links. Liao et al. [16] considered the topological attributes of the network by using multiple characteristics, and rank the nodes in the VN and SN by the product of local resource and these characteristics. Base on the node ranks, three topology-aware VNE algorithms were proposed, which leverage the respective advantages of different characteristics. However, the aforementioned algorithms either typically only considered single topological attribute, or might lead to the imbalance of different resource attributes when evaluating the resource of nodes, which will result in decreased utilization rate of substrate resource and low revenues of InPs.

Our work differs from the existing algorithms in three aspects. First, we consider multiple node attributes together and redefine them for the VNE problem with dynamic states of nodes and links, which reflect the real-time resources of nodes and topological attributes of networks from both the local and global views, respectively. These attributes are leveraged in the node mapping stage and lead to better coordination between the node and link mappings. Second, we take the hops of the substrate paths that virtual links are mapped onto into consideration in the node mapping stage, which can reduce the unnecessary consumption of link bandwidth and increase the utilization rate of substrate resource. Third, we apply TOPSIS to evaluate the importance of nodes based on multiple node attributes, aiming at balancing these attributes when ranking the nodes in the VN and SN. Different from previous work that ranks the nodes simply by the product of different node attributes [13-16], our work mends this gap.

3. Problem Statement

In this section, we first describe the network model and the VNE problem. Then we present the objectives of virtual network embedding.

3.1 Network model

The substrate network is modeled as a weighted undirected graph $G_S = (N_S, L_S)$, where N_S and L_S refer to the set of substrate nodes and links, respectively. For each substrate node $n_s \in N_S$, we consider the typical CPU as its attribute, and $CPU(n_s)$ denotes the available CPU resource that the substrate node n_s can provide. For each substrate link $l_s \in L_S$, we consider the bandwidth as its attribute, and $BW(l_s)$ denotes the available bandwidth resource that the substrate link l_s can provide. An example of substrate network is presented in **Fig. 1**, where the numbers in rectangles next to the nodes represent the amount of available CPU resources and the numbers next to the edges represent the amount of available bandwidth resources. Moreover, we use P_S to denote the set of loop-free substrate paths in the substrate network, and for each substrate path $p \in P_S$, the available bandwidth is the minimum bandwidth that the substrate links in p can provide.

Similarly, we model the virtual network request as a weighted undirected graph $G_V = (N_V, L_V)$, where N_V and L_V refer to the set of virtual nodes and links, respectively. The required CPU resource of each virtual node $n_v \in N_V$ is denoted by $CPU(n_v)$, and the required bandwidth resource of each virtual link $l_v \in L_V$ is denoted by $BW(l_v)$. Two VNRs with node and link constraints are presented in **Fig. 1**, where the numbers in rectangles and the numbers next to the virtual links represent the required CPU and bandwidth resources, respectively.

3.2 Virtual network embedding problem

The VNE problem is defined as a mapping M from G_V to a subset of G_S, such that the constraints of nodes and links in the VNR are satisfied, which can be denoted by $M : G_V \rightarrow (N_S^*, P_S^*, R_S^n, R_S^l)$, where $N_S^* \subseteq N_S$ and $P_S^* \subseteq P_S$, R_S^n and R_S^l represent the resources of substrate nodes and links that allocated to the VNR, respectively.

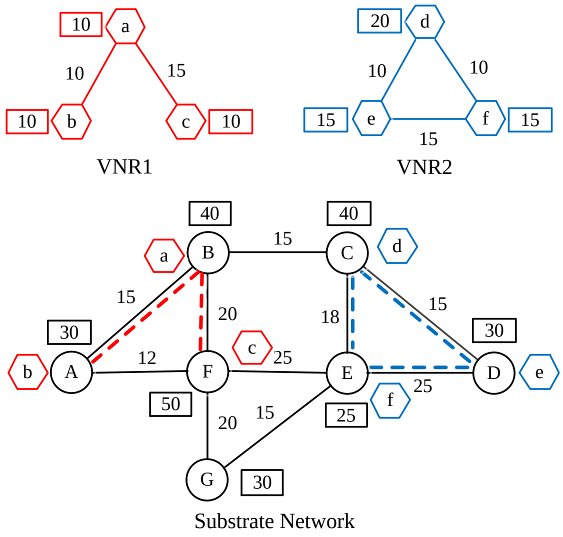


Fig. 1. Example of VNE

The common VNE process, as shown in **Fig. 1**, consists of a node mapping stage and a link mapping stage. We denote the node mapping stage and the link mapping stage by $M_N:(N_V, C_V^n) \to (N_S^*, R_S^n)$ and $M_L:(L_V, C_V^l) \to (P_S^*, R_S^l)$, respectively, where C_V^n and C_V^l represent the required node resources and link resources, respectively. A solution of VNE for VNR1 and VNR2 is illustrated in **Fig. 1**.

3.3 Objectives

In order to reveal the long-term effects of the virtual network embedding, this paper considers the long-term average revenue, the long-term revenue to cost ratio (*R/C*) and the VNR acceptance ratio as the embedding objectives.

Similar to the previous work [9, 10, 11, 18], the revenue of accepting a VNR at time *t* refers to the total resources it demands, which can be formulated as follows:

$$R(VNR,t) = \sum_{n_v \in N_V} CPU(n_v) + \alpha \sum_{l_v \in L_V} BW(l_v) \tag{1}$$

where α is a relative weight to balance the revenue between CPU and bandwidth.

The cost of accepting a VNR at time *t* is the total substrate resources allocated to the VNR, which can be formulated as follows:

$$C(VNR,t) = \sum_{n_v \in N_V} CPU(n_v) + \beta \sum_{l_v \in L_V} BW(l_v) \cdot HOP(l_v) \tag{2}$$

where $HOP(l_v)$ is the number of hops in the substrate path that the virtual link l_v mapped onto, and β is a relative weight to balance the cost between CPU and bandwidth.

Therefore, the long-term average revenue *R* is defined as:

$$R = \lim_{T \to \infty} \frac{1}{T} \sum\nolimits_{t=0}^{T} R(VNR,t) \tag{3}$$

The long-term average cost *C* is defined as:

$$C = \lim_{T \to \infty} \frac{1}{T} \sum\nolimits_{t=0}^{T} C(VNR,t) \tag{4}$$

The long-term revenue to cost ratio *R/C* is defined as:

$$R/C = \lim_{T \to \infty} \frac{\sum\nolimits_{t=0}^{T} R(VNR,t)}{\sum\nolimits_{t=0}^{T} C(VNR,t)} \tag{5}$$

where we can see that the *R/C* refers to the resource utilization rate of the SN. The larger the *R/C* is, the higher the utilization rate of substrate resource is, and thus the VNE algorithm is more efficient.

The VNR acceptance ratio is defined as the number of VNRs successfully accepted by the SN to the number of total arrival VNRs, which can be formulated as follows:

$$\eta_{accept} = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} VNR_{suc}}{\sum_{t=0}^{T} VNR_{sum}} \tag{6}$$

where VNR_{suc} represents the number of VNRs successfully accepted by the SN and VNR_{sum} represents the total arrival VNRs. We can see that the higher the acceptance ratio is, the more VNRs are mapped successfully in a certain period of time, and thus InPs can obtain larger revenues.

4. Multi-attribute Node Ranking

4.1 Motivations

As the basis of node mapping, the node ranking is crucial to the performance of VNE algorithms. It evaluates the embedding potential of each node in the network, so as to correspondingly map them one after another. Previous work in [10] ranks nodes by the product of the CPU resources and the total bandwidth resources of their adjacent links, which can be formulated by Eq. (7), and many VNE algorithms later have adopted such node ranking method [11, 12].

$$H(n) = CPU(n) \sum_{l\in L(n)} BW(l) \tag{7}$$

where $L(n)$ is the set of all adjacent links of node n.

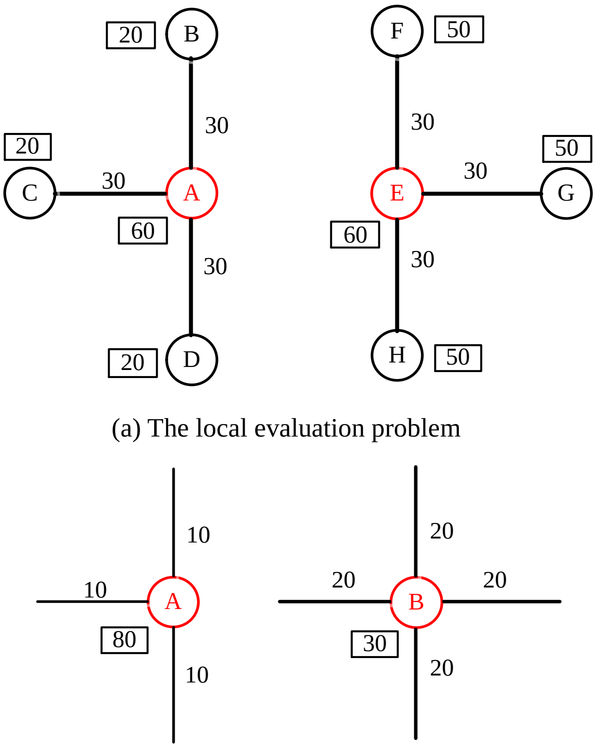
However, such node ranking method has the following disadvantages.

First, it only considers the local resources of nodes and ignores their topological attributes that may pose great effect on the performance of VNE, which can't reflect the real embedding potential of nodes. For example, as shown in **Fig. 2(a)**, where the numbers in rectangles next to the nodes represent CPU resources and the numbers next to the edges represent bandwidth resources. Node A and node E have the same available local resources as $60\times(30+30+30)=5400$. However, the CPU resources of the neighbor nodes of node E are more than those of node A, and the embedding potential of node E should be larger than that of node A.

Second, such method will lead to the imbalance of different node attributes, which will in turn cause the obtained results to be not relatively good in all the node attributes [20]. For example, as shown in **Fig. 2(b)**, $H(\mathrm{A})$ is equal to $H(\mathrm{B})$, but we prefer substrate node B to node A when mapping a virtual node C requiring 20 units of CPU resource. Because node B have more adjacent bandwidth resources, and mapping the virtual node C onto B will have a higher chance to achieve a successful link mapping and increase the utilization rate of resource.

Although the former disadvantage can be overcome by considering additional topological attributes in the node mapping stage [13-16], the latter one has yet been addressed properly.

Network centrality is an important topological attribute, which has been widely used in mining the important nodes in social network analysis. Generally, the closer a node is to the network center, the more important it is. The common centrality criterions in network analysis are degree and closeness, which reflect different topological attributes of nodes and measure the relative importance of nodes from different aspects. Specially, the degree centrality reflects the local importance of a node, and the closeness centrality focuses on the global importance of a node based on the shortest path.



(a) The local evaluation problem

(b) The unbalanced evaluation problem

Fig. 2. Motivational example

In this section, we introduce the degree and closeness centralities into the VNE problem to measure the importance of nodes in the VN and SN. Besides, we also take the resource capacity, communication capacity, and correlation quality as the evaluation criterions of node importance for the VNE problem, where the importance refers to the relative embedding potential of a node. After defining and analyzing the above criterions, we propose a novel node ranking algorithm (TOP-MANR), which ranks the nodes in the VN and SN according to their importance measured by TOPSIS [21].

4.2 Node importance analysis

Different definitions of network centralities will lead to different node ranks. Since the VNs and the SN in the VNE problem are weighted networks, and the state of nodes and links are changed dynamically, rather than simple graph theory, we redefine and extend the degree and closeness centralities to a new format for the VNE problem in this subsection. Besides, we define resource capacity, communication capacity, and correlation quality as the evaluation criterions, which measure the relative importance of nodes from different aspects.

Definition 1 (Resource Capacity, RC) The resource capacity of node n_i is defined as follows:

$$RC(n_i) = CPU(n_i) + \sum_{n_j \in Nb(n_i)} CPU(n_j) \frac{BW(l_{ij})}{\sum_{n_k \in Nb(n_j)} BW(l_{jk})} \tag{8}$$

where $Nb(n_i)$ denotes the set of the neighbor nodes that directly connect the node n_i by a link. If $n_i \in N_V$, $CPU(n_i)$ denotes its required CPU resources, $BW(l_{ij})$ denotes the required bandwidth resources of virtual link l_{ij}. If $n_i \in N_S$, $CPU(n_i)$ refers to its available CPU resources, $BW(l_{ij})$ refers to the available bandwidth resources of substrate link l_{ij}.

As shown in Fig. 2(a), node E is a more important node because the neighbor nodes of node E, namely, F, G, H, have more resources than those of node A's neighbors. Therefore, the resource capacity of node n_i considers both the CPU resources of itself and those of its neighbor nodes. For a virtual node $n_i \in N_V$, the higher its resource capacity is, the larger resources it demands, and it is more difficult to map due to the lack of resources in the SN. So we consider it more important and should be mapped first. For a substrate node $n_i \in N_S$, the higher its resource capacity is, the larger available resources it has, and it is more important in the SN as it can facilitate the node mapping.

Definition 2 (Communication Capacity, CC) The communication capacity of node n_i is defined as the total bandwidth of its adjacent links:

$$CC(n_i) = \sum_{l \in L(n_i)} BW(l) \tag{9}$$

where $L(n_i)$ denotes the adjacent link set of node n_i.

Definition 3 (Degree, D) The degree of node n_i is defined as the number of its adjacent links:

$$D(n_i) = dg(n_i) \tag{10}$$

where $dg(n_i)$ denotes the number of the adjacent links of node n_i.

The degree centrality and communication capacity reflect the local importance of the node in the network. A node with high degree has many link connections while a node with large communication capacity has strong connections. As show in Fig. 2(a), the degrees of node A and node E are both 3, and the communication capacity of node A and node E are both 90.

Definition 4 (Closeness, C) The closeness of node n_i is defined as follows:

$$C(n_i) = \sum_{n_j \in N} \frac{bw(n_i, n_j)}{d(n_i, n_j)} \tag{11}$$

where N denotes the set of nodes, $bw(n_i, n_j)$ denotes the available bandwidth of the shortest path between node n_i and node n_j. $d(n_i, n_j)$ refers to the distance of the shortest path between node n_i and node n_j, which is described by the number of hops along this path, and when $i = j$, $bw(n_i, n_j)/d(n_i, n_j) = 0$.

The traditional closeness centrality is defined by the reciprocal of the sum of shortest distances from one node to all other nodes in the network [13], which only considers the topology of the network. However, since the VN and SN are weighted networks and changed dynamically due to the continuous VNE, we modify the closeness centrality as Eq. (11) for the VNE problem, which takes the bandwidth of links into consideration. Thus, if a node stays near to other nodes in the network and has strong connections between them, the closeness value will be large, and it is more important than other nodes in the network and will be helpful to increase the utilization rate of resource.

Definition 5 (Correlation quality, CQ) The correlation quality of substrate node n_s is defined as follows:

$$CQ(n_s) = \sum_{n_i \in N_V^M} e^{\frac{bw(M_N(n_i), n_s)}{hops(M_N(n_i), n_s)}} \tag{12}$$

where N_V^M denotes the set of virtual nodes that have been mapped onto the SN, $M_N(n_i)$ denotes the substrate node that the virtual node n_i mapped onto, $hops(M_N(n_i), n_s)$ refers to the hops of the shortest path between node $M_N(n_i)$ and node n_s .

From Eq. (12), we know that the virtual nodes in N_V^M have been mapped onto the SN. If an unmapped virtual node $n_v \in N_V$ is mapped onto the substrate node n_s that is far away from $M_N(n_i)$ ($n_i \in N_V^M$), the cost of mapping virtual links between n_v and $n_i \in N_V^M$ will be too large according to Eq. (2), resulting in low resource utilization rate of the SN. Therefore, to reduce the hops of substrate paths in the link mapping stage, CQ considers both the available bandwidth resources and hops of substrate paths in the node mapping stage, which will keep the selected substrate nodes that the virtual nodes are mapped onto connected closely to each other and facilitate the subsequent link mapping. So the substrate node with larger CQ is more important in the SN and should be selected as the mapped node in priority.

4.3 TOP-MANR algorithm

In the node mapping stage, the more important virtual node has higher mapping priority, and should be mapped onto the substrate node with higher importance. The above five criterions of node importance for the VNE problem reflect either the local importance or the global importance of a node in the network. However, it is one-sided to determine the importance of nodes only relying on one of the criterions in real network. Therefore, different from previous work in [13-16], which measure the node importance simply by the product of different resource attributes, we combine these criterions together and propose a multi-attribute node ranking algorithm based on TOPSIS (TOP-MANR), aiming at achieving a balanced importance evaluation results and selecting the nodes with relatively good performance in all node attributes.

TOPSIS is a known classical method for dealing with multi-attribute decision making problems [21]. It bases upon the principle that the chosen solution should have the shortest distance from the positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS). In this subsection, we regard each node in the VN or SN as a solution, and each evaluation criterion of node importance as a solution's attribute, and then the evaluation of node importance is transformed into a multi-attribute decision making problem.

The procedures of the TOP-MANR algorithm are as follows.

Step 1: Consider a network with *F* nodes, each node has *M* importance evaluation criterions, and the criterion *j* of node *i* is denoted by x_{ij}. Thus, the importance decision matrix can be expressed as follows:

$$\boldsymbol{X}_{F\times M}=\begin{bmatrix} x_{11} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{F1} & \cdots & x_{FM} \end{bmatrix} \tag{13}$$

Step 2: Since different criterions have different dimensions and orientation of merits, we normalize the initial data and eliminate the impact of dimension by dimensionless treatment for comparison, and then we obtain the normalized decision matrix:

$$\boldsymbol{X}^{'}_{F\times M}=\begin{bmatrix} x^{'}_{11} & \cdots & x^{'}_{1M} \\ \vdots & \ddots & \vdots \\ x^{'}_{F1} & \cdots & x^{'}_{FM} \end{bmatrix} \tag{14}$$

where for the positive criterion, namely bigger is better, $x^{'}_{ij}=(x_{ij}-\min\limits_{1\le k\le F}x_{kj})\Big/(\max\limits_{1\le k\le F}x_{kj}-\min\limits_{1\le k\le F}x_{kj})$, and for the negative criterion, namely smaller is better, $x^{'}_{ij}=(\max\limits_{1\le k\le F}x_{kj}-x_{ij})\Big/(\max\limits_{1\le k\le F}x_{kj}-\min\limits_{1\le k\le N}x_{kj})$.

Step 3: Considering the different significance of each criterion, we denote the weight of the criterion *j* as ω_j $(j=1,2,\ldots,M,0\le\omega_j\le 1,\sum_{j=1}^{M}\omega_j=1)$, then the weighted normalized decision matrix can be expressed as:

$$\boldsymbol{X}^{''}_{F\times M}=\begin{bmatrix} x^{''}_{11} & \cdots & x^{''}_{1M} \\ \vdots & \ddots & \vdots \\ x^{''}_{F1} & \cdots & x^{''}_{FM} \end{bmatrix} \tag{15}$$

where $x^{''}_{ij}=\omega_j x^{'}_{ij}$.

Step 4: Denote the positive ideal solution as A^{+} (PIS) and the negative ideal solution as A^{-} (NIS), which are formulated as follows:

$$A^{+}=\{x^{+}_{1},x^{+}_{2},\ldots,x^{+}_{M}\}=\{\max_{1\le i\le F}x^{''}_{i1},\max_{1\le i\le F}x^{''}_{i2},\ldots,\max_{1\le i\le F}x^{''}_{iM}\} \tag{16}$$

$$A^{-}=\{x^{-}_{1},x^{-}_{2},\ldots,x^{-}_{M}\}=\{\min_{1\le i\le F}x^{''}_{i1},\min_{1\le i\le F}x^{''}_{i2},\ldots,\min_{1\le i\le F}x^{''}_{iM}\} \tag{17}$$

Step 5: Calculate the Euclidean distances of each solution to the PIS A^{+} and the NIS A^{-}, which is denoted by D^{+}_{i} and D^{-}_{i}, respectively, and formulated as follows:

$$D_i^+ = \sqrt{\sum_{j=1}^{M}(x_{ij}^{"} - x_j^+)^2}, \quad i = 1, 2, ..., F \tag{18}$$

$$D_i^- = \sqrt{\sum_{j=1}^{M}(x_{ij}^{"} - x_j^-)^2}, \quad i = 1, 2, ..., F \tag{19}$$

Step 6: Calculate the relative closeness C_i of each solution to the ideal solutions, which can be expressed as follows:

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}, \quad i = 1, 2, ..., F \tag{20}$$

where $0 \le C_i \le 1$. The larger the index value is, the more important the node is in the network.

Step 7: Rank the nodes in the network according to the descending order of the value of C_i.

In this paper, we take the resource capacity, the communication capacity, the degree and the closeness as the evaluation criterions of importance for virtual nodes and substrate nodes. Besides, the correlation quality is also selected as an evaluation criterion of importance for substrate nodes, which ensures the hops of the substrate paths that the virtual links are mapped onto are not too large. For all these five criterions, a larger value is better. Then, we can obtain the ranking sequences for virtual nodes and substrate nodes through the TOP-MANR algorithm.

5. Heuristic Algorithm Design

Based on the multi-attribute node ranking, we propose a novel heuristic algorithm (TOP-VNE) for the VNE problem, which consists of the node mapping stage based on TOP-MANR and the link mapping stage based on the shortest-path algorithm.

5.1 Node mapping algorithm

In the node mapping stage, we adopt a greedy strategy to map the virtual nodes in the VNR onto the substrate nodes in the SN according to their ranks, which helps to satisfy the resource requirements of the current VNR and balance the loads of the substrate network.

The details of the node mapping algorithm are shown in Algorithm 1, and it works as follows. After calculating the RC, CC, D and C of nodes in the VNR, we sort them by the TOP-MANR algorithm. For the virtual node with the highest rank that is unmapped in the VNR, we build a set of candidate substrate node, consisting of nodes whose available CPU resources can satisfy the requirements of the virtual node and are unmapped with any other nodes in the same VNR. If there are no candidate substrate nodes for the virtual node, the node mapping is failed. Otherwise, we calculate the RC, CC, D, C and CQ of each node in the candidate set and sort them by the TOP-MANR algorithm. Finally, we map the virtual node onto the candidate substrate node with the highest rank. Other virtual nodes in the same VNR will be mapped in the same way. Note that the RC, CC, C and CQ of the substrate node are all calculated using its real-time available resource, and the states of the substrate network are changed dynamically during the node mapping stage. Thus the evaluation criterions of virtual nodes and substrate nodes shouldn't be calculated at the same time.

Algorithm 1 Node Mapping Algorithm

```
input: $G_S = (N_S, L_S)$ : substrate network
       $G_V = (N_V, L_V)$ : the arriving VNR
Output: $M_N$ : node mapping
1. for each virtual node $n_v \in N_V$ do
2.   Calculate $RC(n_v)$, $CC(n_v)$, $D(n_v)$ and $C(n_v)$;
3. end for
4. Rank the virtual nodes in current VNR by the TOP-MANR algorithm;
5. for all the unmapped virtual nodes in current VNR do
6.   Choose the virtual node $n_v$ with the highest rank;
7.   Construct the set of candidate substrate nodes $\Omega(n_v)$ for $n_v$;
8.   if $\Omega(n_v) = \Phi$ then
9.     Return NODE_MAPPING_FAILED;
10.  else
11.    for each candidate node $n_s$ in $\Omega(n_v)$ do
12.      Calculate $RC(n_s)$, $CC(n_s)$, $D(n_s)$, $C(n_s)$ and $CQ(n_s)$;
13.    end for
14.      Rank the candidate nodes in $\Omega(n_v)$ by the TOP-MANR algorithm, and map $n_v$ onto the
         candidate node $n_s$ with the highest rank, namely $M_N(n_v) = n_s$;
15.  end if
16. end for
17. Return NODE_MAPPING_SUCCESS;
```

5.2 Link mapping algorithm

In the link mapping stage, TOP-VNE adopts the shortest-path based algorithm to map virtual links onto the shortest substrate paths. Since different substrate paths that virtual links are mapped onto may share the same substrate links and compete for their limited bandwidth resources, it is difficult or even impossible to map virtual links with large bandwidth demands due to the lack of bandwidth resources in the SN. Therefore, virtual links with large bandwidth demands should be mapped in priority. Similar to the previous work [19], for each virtual link in the VNR, we pre-cut all the links in the SN that do not satisfy the bandwidth requirements to make the link mapping more efficient. Then we adopt the Dijkstra's algorithm to compute the shortest path between the corresponding substrate nodes. If no shortest path exists, the link mapping is failed. Algorithm 2 gives the details of the link mapping algorithm.

5.3 Time complexity analysis

The time complexity of TOP-MANR algorithm mainly depends on the computing of closeness centrality. For a network with m nodes, the time complexity of Dijkstra's algorithm to compute the shortest path between two nodes is $O(m^2)$, so the time complexity of ranking nodes in the network by TOP-MANR is $O(m^3)$.

TOP-VNE is a two-stage algorithm. The complexity of ranking nodes in VNR is $O(|N_V|^3)$, and for each virtual node, the complexity of ranking its substrate candidate nodes is $O(|N_S|^3)$, so the time complexity of Algorithm 1 is $O(|N_V|^3+|N_V||N_S|^3)$. For each virtual link, the complexity of calculating its shortest path is $O(|N_S|^2)$, so the time complexity of Algorithm 2 is $O(|L_V||N_S|^2)$. Therefore, the time complexity of TOP-VNE algorithm is $O(|N_V|^3+|N_V||N_S|^3+|L_V||N_S|^2)$, and it can be solved in polynomial time.

Algorithm 2 Link Mapping Algorithm

```
Input: $G_S=(N_S,L_S)$: substrate network
       $G_V=(N_V,L_V)$: the arriving VNR
       $M_N$: node mapping
Output: $M_L$: link mapping
1. Rank the virtual links in current VNR according to the required bandwidth in descending order;
2. for all the unmapped virtual links in current VNR do
3.   Choose the virtual link $l_{uv}$ with the highest rank;
4.   $G_S^{temp} \leftarrow G_S$;
5.   for each substrate link $l_{ij}$ in $G_S^{temp}$ do
6.     if $BW(l_{ij}) \le BW(l_{uv})$ then
7.       cut $l_{ij}$ in $G_S^{temp}$;
8.     end if
9.   end for
10.  Calculate the shortest path $p_{uv}$ from $M_N(n_u)$ to $M_N(n_v)$ in $G_S^{temp}$ by Dijkstra's algorithm;
11.  if no shortest path exists then
12.    Return LINK_MAPPING_FAILED;
13.  else
14.    $M_L(l_{uv})=p_{uv}$;
15.  end if
16. end for
17. Return LINK_MAPPING_SUCCESS;
```

6. Performance Evaluation

In this section, we first describe the simulation environment, and then present our main evaluation results. Our evaluation focuses primarily on the comparison of TOP-VNE with several existing VNE algorithms in terms of long-term average revenue, long-term *R/C* ratio, VNR acceptance ratio and runtime.

6.1 Simulation environment

Similar to the previous work [10], we use GT-ITM tool [22] to generate the topologies of SN and VNRs. The SN is configured with 100 nodes and about 500 links. The CPU resources of substrate nodes and the bandwidth resources of substrate links are real numbers uniformly distributed between 50 and 100. We assume that the arrival of VNRs follows the Poisson process with an average arrival rate of 5 VNRs per 100 time units, and each VNR has an exponentially distributed lifetime with an average of 1000 time units. In each VNR, the number of virtual nodes is uniformly distributed between 2 and θ ($\theta > 2$), and the link connectivity rate of virtual node pair is set as η ($0 < \eta \le 1$). The CPU and bandwidth resources requirements of virtual nodes and links are real numbers uniformly distributed between 0 and 50. Besides, for the virtual nodes, we set the weight of the four importance evaluation criterions as $\omega_i = 1/4$ $(i = 1,2,3,4)$, and for the substrate nodes, we set the weight of the five importance evaluation criterions as $\omega_i = 1/5 \ (i = 1,2,3,4,5)$.

Since VNRs are randomly generated, we run our simulation in each condition for 50000 time units to achieve a stable-state performance. Each simulation is performed ten instances and we record the average value as the final result.

Several metrics are used to evaluate the performance of the VNE algorithms, including long-term average revenue (Eq. (3)), long-term *R/C* ratio (Eq. (5)), VNR acceptance ratio (Eq. (6)) and runtime of the algorithm. Our simulation experiments evaluate four algorithms listed in **Table 1**. TOP-VNE is the algorithm we have proposed in this paper, G-SP is the classical greedy algorithm proposed in [10], IC-SP [13] and RW-MM-SP [11] consider the local resources and topological attributes in the node ranking, but they don't balance different factors.

Table 1. Algorithms comparison

Notation	Description
TOP-VNE	The proposed algorithm using TOP-MANR to rank the nodes in the node mapping stage
G-SP	Greedy node mapping that ranks the nodes based on the local resources with shortest path based link mapping
IC-SP	Greedy node mapping that ranks the nodes based on the improved closeness with shortest path based link mapping
RW-MM-SP	Greedy node mapping that ranks the nodes based on Markov Random Walk model with shortest path based link mapping

6.2 Evaluation results

6.2.1 Evaluation for general VNRs

We first evaluate the performance of the aforementioned four VNE algorithms by setting a fixed maximum number of virtual nodes θ as 10 and a fixed link connectivity rate η as 0.5 for the VNRs. The key observations from our simulations are summarized as follows.

Fig. 3 shows the long-term average revenue of the four VNE algorithms in stable state. It can be seen that our TOP-VNE generates the largest average revenue. The reason is that TOP-VNE considers minimizing the hops of the substrate paths that the virtual links are mapped onto, so it can reduce the substrate resources allocated to VNRs. As a result, it can accept more VNRs in a certain period of time and produce larger revenues. Besides, TOP-VNE adopts multi-attribute node ranking, which leads to its largest average revenue.

Fig. 4 shows the long-term *R/C* ratio of the four VNE algorithms in stable state. It can be seen that the *R/C* ratio of our TOP-VNE is higher than the other three VNE algorithms. Moreover, the *R/C* ratio of TOP-VNE is almost 0.755, which is almost 18.7% higher than IC-SP, 22.8% higher than RW-MM-SP, and 30.4% higher than G-SP. According to the definition of *R/C* ratio, we know that the cost of mapping virtual links have significant effect on it. Thus, TOP-VNE achieves higher *R/C* ratio because it considers mapping the virtual nodes onto the substrate nodes nearby, and as a result, the hops of substrate paths that the virtual links are mapped onto are reduced, leading to a low cost of link mapping and high *R/C* ratio.

Fig. 5 shows the acceptance ratio of the four VNE algorithms in stable state. It can be seen that our TOP-VNE outperforms the other three VNE algorithms. Moreover, the acceptance ratio of TOP-VNE is almost 0.929, which is almost 5.5% higher than IC-SP, 18.2% higher than RW-MM-SP, and 22.7% higher than G-SP. The reason is that TOP-VNE considers multiple topological characteristics together and balances them when ranking the nodes, which better coordinates the node mapping stage and the link mapping stage, and results in its acceptance ratio being the highest.

Fig. 6 shows the average runtime of the four algorithms run on the same PC. It can be seen that the runtime of our TOP-VNE is slightly larger than RW-MM-SP and IC-SP. The main reason is that TOP-VNE need to compute the node ranks both in VNRs and SN in the node mapping stage, which leads to a larger runtime. However, TOP-VNE is also a polynomial time algorithm, and one can conclude that the differences in *R/C* ratio, revenue and acceptance ratio justify the affordable runtime spent by TOP-VNE.

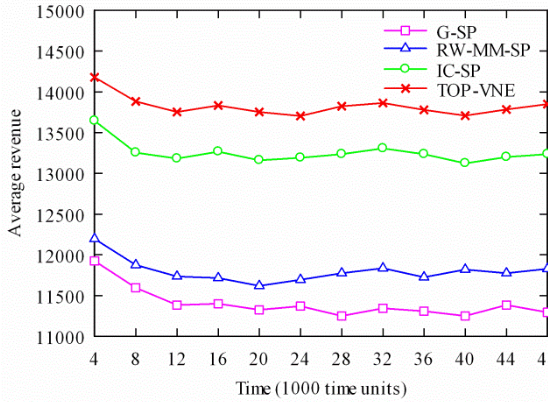


Fig. 3. Average revenue in stable state

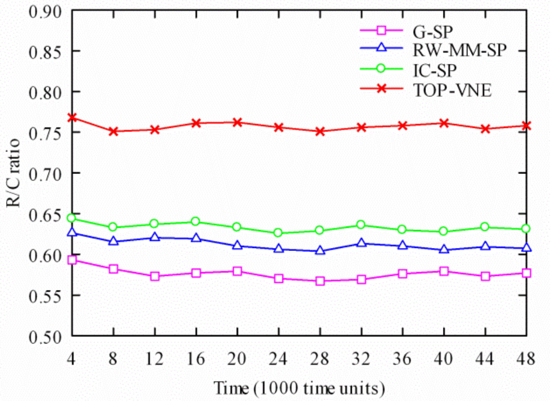


Fig. 4. *R/C* ratio in stable state

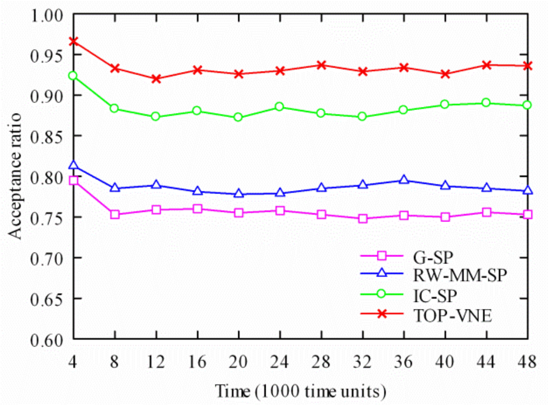


Fig. 5. Acceptance ratio in stable state

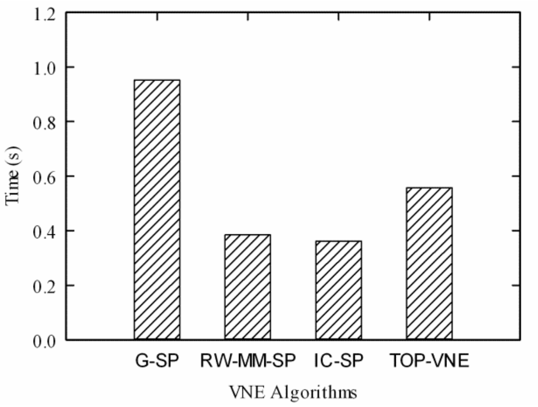


Fig. 6. Runtime in stable state

6.2.2 Evaluation for different kinds of VNRs

To further evaluate the performance of our TOP-VNE, we compare the *R/C* ratio and the acceptance ratio of the four VNE algorithms under different numbers of virtual nodes and link connectivity rates for VNRs.

First, we set the maximum number of virtual nodes θ as increasing from 5 to 40 while keeping a fixed link connectivity rate η at 0.5.

Figs. 7 and **8** show the long-term *R/C* ratio and the acceptance ratio of four VNE algorithms in stable state, respectively. From the figures, we can see that the *R/C* ratio and the acceptance ratio decrease as the number of virtual nodes increases. But our TOP-VNE still have a relative higher *R/C* ratio (IC-SP: 21.7% on average, RW-MM-SP: 28.4% on average, G-SP: 42.1% on average) and a relative higher acceptance ratio (IC-SP: 7.7% on average, RW-MM-SP: 19.2% on average, G-SP: 28.2% on average) than the other three VNE algorithms. It can be understood as follows. As the number of virtual nodes increases, the number of virtual links connecting each pair of virtual nodes also increases. As a result, VNRs need to consume more resources if accepted, resulting in a lower *R/C* ratio and acceptance ratio. The evaluation results indicate that TOP-VNE is more efficient when the VNRs require more virtual nodes.

Second, we set the link connectivity rate of VNRs as increasing from 0.1 to 1.0 while keeping a fixed maximum number of virtual nodes θ at 10.

Figs. 9 and **10** show the long-term *R/C* ratio and the acceptance ratio of four VNE algorithms in stable state, respectively. From the figures, we can see that the *R/C* ratio and the acceptance ratio decrease as the link connectivity rate increases. But our TOP-VNE still have a relative higher *R/C* ratio (IC-SP: 14.8% on average, RW-MM-SP: 26.7% on average, G-SP: 31.5% on average) and a relative higher acceptance ratio (IC-SP: 8.2% on average, RW-MM-SP: 21.3% on average, G-SP: 28.6% on average) than the other three VNE algorithms. It can be understood as follows. As the link connectivity rate increases, the number of virtual links required by VNRs also increases. Thus, VNRs need to consume more bandwidth resources in VNE, leading to a lower *R/C* ratio and acceptance ratio. The evaluation results demonstrate that our TOP-VNE achieves a much better performance under different link connectivity rates condition. Compared with **Figs. 7** and **8**, we note that the curves in **Figs. 7** and **8** drop down more steeply than that in **Figs. 9** and **10**, which reflects that the number of virtual nodes have greater influence on the performance of the VNE algorithms than the link connectivity rate.

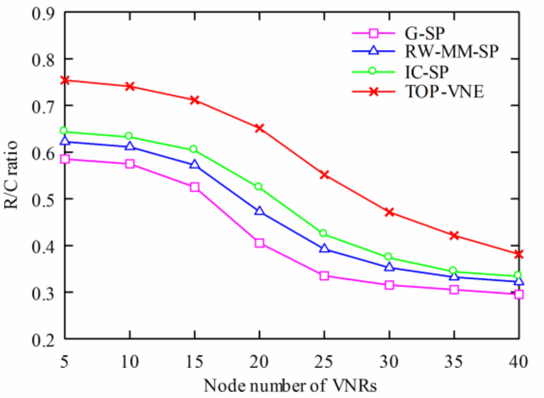


Fig. 7. *R/C* ratio under different number of virtual nodes in stable state

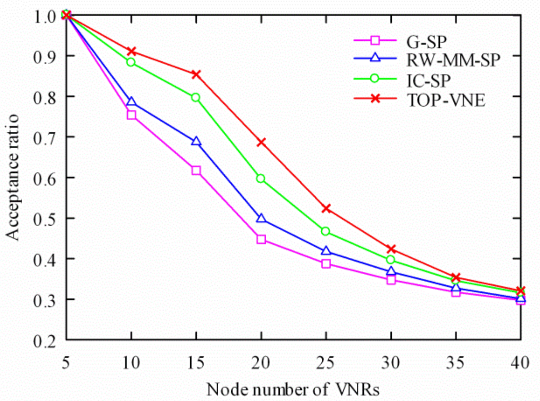


Fig. 8. Acceptance ratio under different number of virtual nodes in stable state

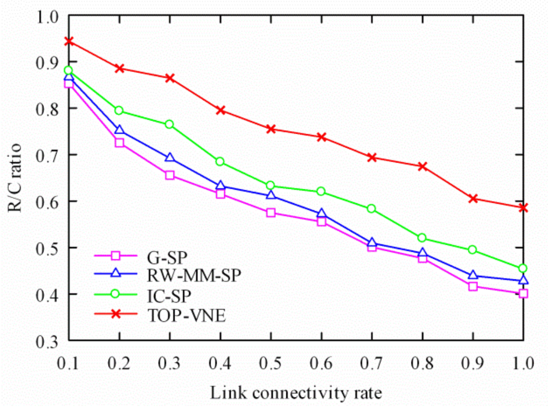


Fig. 9. *R/C* ratio under different link connectivity rate in stable state

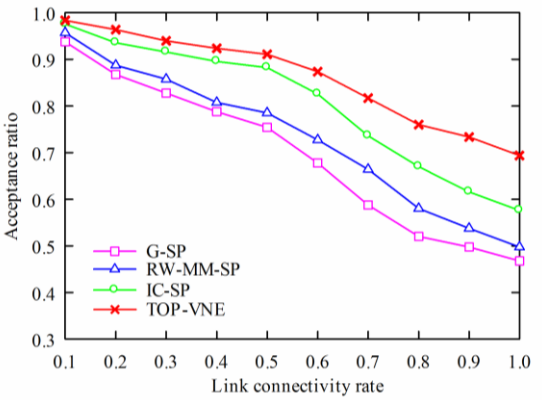


Fig. 10. Acceptance ratio under different link connectivity rate in stable state

7. Conclusion

Virtual network embedding is one of the main challenges in network virtualization. In this paper, we introduce and redefine five node attributes to analyze the embedding potential of nodes in the VNs and the SN, and then rank them according to their importance in the network evaluated by using TOPSIS method. Based on the node ranks, we propose a novel VNE algorithm TOP-VNE. Simulation results show that the proposed algorithm outperforms the existing VNE algorithms in terms of the long-term average revenue, the long-term R/C ratio and the VNR acceptance ratio under different conditions of VNRs.

In the future work, we will take more topological attributes (e.g. eigenvector centrality, betweenness centrality) into consideration to improve the performance of the algorithm. Besides, we will extend our work by considering more aspects of VNE problems, e.g. energy consumption, fault tolerance, security issues.

References

[1] T. Anderson, L. Peterson, S. Shenker and J. Turner, "Overcoming the Internet impasse through virtualization," *IEEE Computer Magazine*, vol. 38, no. 4, pp. 34-41, April, 2005. Article (CrossRef Link).

[2] N. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862-876, May, 2010. Article (CrossRef Link).

[3] Ashiq Khan, Alf Zugenmaier, Dan Jurca and Wolfgang Kellerer, "Network virtualization: a hypervisor for the Internet?" *IEEE Communication Magazine*, vol. 50, no. 1, pp. 136-143, January, 2012. Article (CrossRef Link).

[4] A.J. Wang, M. Iyer, R. Dutta, G.N. Rouskas and I. Baldine, "Network virtualization: technologies, perspectives, and frontiers," *Journal of Lightware Technology*, vol. 31, no. 4, pp. 523-537, February, 2013. Article (CrossRef Link).

[5] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer and X. Hesselbach, "Virtual network embedding: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888-1906, Fourth Quarter, 2013. Article (CrossRef Link).

[6] D.G. Andersen, "Theoretical approaches to node assignment," http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps, 2002.

[7] I. Fajjari, N. Aitsaadi, M. Pióro and G. Pujolle, "A new virtual network static embedding strategy within the cloud's private backbone network," *Computer Networks*, vol. 62, no. 5, pp. 69-88, May, 2014. Article (CrossRef Link).

[8] D. Liao, G. Sun, V. Anand and H, Yu, "Efficient provisioning for multicast virtual network under single regional failure in cloud-based datacenters," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 7, pp. 2325-2349, July, 2014. Article (CrossRef Link).

[9] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc. of IEEE INFOCOM*, April 23-29, 2006. Article (CrossRef Link).

[10] M. Yu, Y. Yi, J. Rexford and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17-29, April, 2008. Article (CrossRef Link).

[11] X. Cheng, S. Su, F. Yang et al., "Virtual network embedding through topology-Aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 38-47, April, 2011. Article (CrossRef Link).

[12] J. Liao, S. Qing, J.Y. Wang, X. Zhu and J. Wang, "Hybrid virtual network embedding with time-oriented scheduling policy," *Chinese Journal of Electronics*, vol. 22, no. 4, pp. 789-794, April, 2013.

[13] Z. Wang, Y. Han, T. Lin, H. Tang and S. Ci, "Virtual network embedding by exploiting

topological information," in *Proc. of IEEE GLOBECOM*, pp. 2603-2608, December 3-7, 2012. Article (CrossRef Link).

[14] H. Cui, S. Tang, X. Huang, J. Chen and Y. Liu, "A novel method of virtual network embedding based on topology convergence-degree," in *Proc. of IEEE ICC*, pp. 246-250, June 9-13, 2013. Article (CrossRef Link).

[15] J. Ding, T. Huang, J. Liu and Y. Liu, "Virtual network embedding based on real-time topological attributes," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 2, pp. 109-118, February, 2015. Article (CrossRef Link).

[16] J. Liao, M. Feng, T. Li, J, Wang and S. Qing, "Topology-aware virtual network embedding using multiple characteristics," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 1, pp. 145-164, January, 2014. Article (CrossRef Link).

[17] B. Lu, J. Chen, H. Cui, T. Huang and Y. Liu, "A virtual network mapping algorithm based on integer programming," *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, vol. 14, no. 12, pp. 899-908, December, 2013. Article (CrossRef Link).

[18] M. Chowdhury, M.R. Rahman and R. Boutaba, "ViNEYard: virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206-219, February, 2012. Article (CrossRef Link).

[19] L. Gong, Y. Wen, Z. Zhu and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. of IEEE INFOCOM*, April 27-May 2, 2014. Article (CrossRef Link).

[20] X. Li, H. Wang, B. Ding, X.Y. Li and D. Feng, "Resource allocation with multi-factor node ranking in data center networks," *Future Generation Computer Systems*, vol. 32, no. 2, pp. 1-12, February, 2014. Article (CrossRef Link).

[21] C.L. Hwang and K.P. Yoon, "Multiple attribute decision making: methods and applications," Springer-Verlag, Berlin, Germany, 1981.

[22] E. Zegura, K. Calvert and S. Bhattacharjee, "How to model an Internetwork," in *Proc. of IEEE INFOCOM*, March 24-28, 1996. Article (CrossRef Link).

Shuiqing Gong is a Ph.D. candidate in Computer Application Technology from Air Force Engineering University, China. His research interests include network virtualiztion, software defined networking and cloud computing.

Jing Chen is a professor at Air Force Engineering University, China. She received her B.S. degree in Computer Science and Technology in 1985 from Air Force Engineering University, China, her M.S. degree in Computer Science and Technology in 1994 from Northwestern Polytechnical University, China, and her Ph.D. degree in Communication and Information System in 2005 from Xi'an Jiaotong University, China. From 2011 to 2012, she was a visiting scholar at the Department of Computer Science, Ohio State University. Her research interests include next generation Internet, cloud computing and software defined networking.

Siyi Zhao is a postgraduate in Computer Science from Air Force Engineering University, China. Her research interests include network virtualiztion and software defined networking.

QingChao Zhu is a Ph.D. candidate in Computer Application Technology from Air Force Engineering University, China. His research interests include network virtualiztion and cloud computing.