

# Cognitive Virtual Network Embedding Algorithm Based on Weighted Relative Entropy

Yuze Su, Xiangru Meng, Zhiyuan Zhao and Zhentao Li

College of Information and Navigation, Air Force Engineering University

Xi'an, Shaanxi 710077 - China

[e-mail: glgiuip@163.com]

\*Corresponding author: Yuze Su

*Received July 28, 2017; revised February 5, 2018; revised October 15, 2018; accepted November 7, 2018;  
published April 30, 2019*

---

## Abstract

Current Internet is designed by lots of service providers with different objects and policies which make the direct deployment of radically new architecture and protocols on Internet nearly impossible without reaching a consensus among almost all of them. Network virtualization is proposed to fend off this ossification of Internet architecture and add diversity to the future Internet. As an important part of network virtualization, virtual network embedding (VNE) problem has received more and more attention.

In order to solve the problems of large embedding cost, low acceptance ratio (AR) and environmental adaptability in VNE algorithms, cognitive method is introduced to improve the adaptability to the changing environment and a cognitive virtual network embedding algorithm based on weighted relative entropy (WRE-CVNE) is proposed in this paper. At first, the weighted relative entropy (WRE) method is proposed to select the suitable substrate nodes and paths in VNE. In WRE method, the ranking indicators and their weighting coefficients are selected to calculate the node importance and path importance. It is the basic of the WRE-CVNE. In virtual node embedding stage, the WRE method and breadth first search (BFS) algorithm are both used, and the node proximity is introduced into substrate node ranking to achieve the joint topology awareness. Finally, in virtual link embedding stage, the CPU resource balance degree, bandwidth resource balance degree and path hop counts are taken into account. The path importance is calculated based on the WRE method and the suitable substrate path is selected to reduce the resource fragmentation. Simulation results show that the proposed algorithm can significantly improve AR and the long-term average revenue to cost ratio (LTAR/CR) by adjusting the weighting coefficients in VNE stage according to the network environment. We also analyze the impact of weighting coefficient on the performance of the WRE-CVNE. In addition, the adaptability of the WRE-CVNE is researched in three different scenarios and the effectiveness and efficiency of the WRE-CVNE are demonstrated.

---

**Keywords:** Virtual Network Embedding, Cognitive, Weighted Relative Entropy, Joint Topology Awareness, Breadth First Search

## 1. Introduction

With the rapid development of Internet technology, network ossification has become the key factor restricting the innovation and development of Internet architecture [1]. As a promising technology in next generation network, network virtualization technology allows the substrate network (SN) to be shared by multiple virtual networks (VNs) and provides different kinds of services [2, 3]. Meanwhile, it has been widely used in research testbeds, such as PlanetLab [4], the global environment for network innovations (GENI) [5], virtual network infrastructure (VINI) [6] and federated E-infrastructure dedicated to European researchers innovating in computing network architectures (FEDERICA) [7].

As the main challenge of network virtualization, the VNE deals with the allocation of virtual resources both in nodes and links. It can be divided into virtual node embedding and virtual link embedding. The VNE problem has been proved to be NP-hard [8].

Nowadays, the VNE has gradually become a research hotspot [9-14, 18-29]. In [9], the VNE is systematically summarized and it contains one-stage [10, 11] and two-stage VNE algorithms [12-14, 18-29]. In one-stage VNE algorithms, virtual links are embedded at the same time as virtual nodes. When the virtual node is embedded, the virtual links connecting it are also embedded. The topological attributes of the nodes are taken into consideration and the topology correlation is always perfect. Therefore, the efficiency of the VNE is improved. However, this algorithm is complex and takes more time to run.

In two-stage VNE algorithms, all virtual nodes are embedded first. When the virtual node embedding is completed, the virtual links begin to embed. The lack of coordination between virtual node embedding and virtual link embedding will result in the neighboring virtual nodes being actually widely separated in the substrate topology. This increases the cost in virtual link embedding which results in the low AR and high embedding cost. However, the two-stage VNE algorithms are easy to implement and they are also widely applied. In two-stage VNE algorithms, virtual node embedding as the important stage of VNE has gradually got much attention. The virtual node ranking indicators have great influence on the performance of VNE. The node CPU resource, bandwidth resource [12], network centrality [13] and regional resource clustering coefficient [14] are usually selected as the virtual node ranking indicators. They gradually develop from resource indicators to comprehensive indicators which take into account both the resource factor and topology factor. However, all these algorithms have low environmental adaptability.

In recent years, cognitive method is widely used in some areas of network, such as wireless mesh networks [15, 16] and smart grid [17] to improve the adaptability to the changing environment. As one type of the cognitive network, the cognitive VN is different from traditional networks. It can plan, decide and act adaptively by sensing the current environment, and also learn from adaptive actions. In addition, cognitive method has been gradually used in the process of virtual node embedding, such as topology-aware VNE [11,18], trust-aware secure VNE [19], time and energy aware VNE [20], node adjacent-awareness VNE [22]. However, virtual link embedding algorithm in above VNE algorithms is  $k$ -shortest path algorithm or denoted by one fixed formula.

In summary, the studies about virtual node embedding are focused on two points: one is the selection of node ranking indicators, which has expanded from pure resource or topology indicators to comprehensive indicators. The other one is the ranking strategy of multiple node ranking indicators. Same node ranking indicators combined in different ways have different

impacts on the performance of VNE. However, there are three obvious disadvantages in existing virtual node embedding algorithms. At first, although cognitive method has been introduced into some VNE algorithms, most virtual node embedding strategies are simply fixed by formulas and they are hard to add other indicators. Next, they are only on the view of single virtual or substrate topology awareness and failed to achieve joint awareness. At last, the node ranking indicators and strategies are always same in both VN and SN. There are some obvious differences between SN and VN in size, topology and resources. Therefore, the ranking indicators and strategies of virtual nodes and substrate nodes should be selected according to their own environments. Also, the  $k$ -shortest path algorithm which is widely used in virtual link embedding can't adapt to the changing environment.

To solve above problems, a WRE-CVNE is proposed. First, a WRE method is proposed as the basic of WRE-CVNE. As a multiple indicator ranking algorithm, the ranking indicators and their weighting coefficients are important to the final ranking results. In different environments, multiple indicators can be added or deleted and their corresponding coefficients can be changed with the help of WRE method. Hence, the WRE method improves the adaptability of the WRE-CVNE. Then the node proximity in substrate node ranking and the BFS algorithm in virtual node ranking are used at the same time, which realize the combination of virtual and substrate topology awareness. At last, the WRE method is introduced into VNE which is useful to select the suitable substrate nodes and paths. In addition, three experiments are designed. The performance of the WRE-CVNE is evaluated by comparing with other algorithms first. The second experiment verifies that the weighting coefficients of ranking indicators in virtual node embedding stage have huge impact on the performance of the WRE-CVNE. In the last experiment, the adaptability of the WRE-CVNE to different environments is researched by adjusting the weighting coefficients in VNE.

The main contributions can be summarized as follows. (1) We propose a WRE method as the basic and select the proper indicators in both virtual node and link embedding stages. (2) We introduce the node proximity into substrate node ranking and BFS algorithm into virtual node ranking which realize the joint awareness of virtual topology and substrate topology. (3) We introduce the WRE method into virtual link embedding algorithm to reduce the resource fragmentation. (4) We set three scenarios to verify the adaptability of the WRE-CVNE to different environments.

The rest of paper is organized as follows. We discuss the related work in section 2. Section 3 gives the model of VNE and evaluation indicators. Section 4 presents the WRE method. Section 5 proposes our novel WRE-CVNE. Section 6 describes the performance evaluation and analysis. The paper is concluded in section 7.

## 2. Related Work

The VNE problem deals with allocating the limited resources in SN to the VN requests. In many previous studies, the meta-heuristic algorithms are introduced to balance the embedding performance and computational complexity. For example, Wang et al. in [23] introduce a VNE algorithm with discrete particle swarm optimization (DPSO), which modifies PSO into DPSO to solve the VNE problem. Zhu et al. in [24] present a VNE algorithm based on the artificial fish swarm algorithm. The artificial fish swarm algorithm is introduced to achieve the approximate optimal embedding. Zhu et al. in [25] propose a binary combinatorial optimization model and a VNE algorithm based on artificial bee colony algorithm to reduce the costs of SN. Chang et al. in [26] present a modified discrete particle swarm optimization algorithm (M\_DPSO) to solve the VNE problem. In the M\_DPSO, the particle evolves more

directionally, and the mutually exclusive factor of different particle positions is introduced to resolve the problem of premature and easily converging to local optimal solution.

Moreover, some heuristic VNE algorithms get more and more attention and they are introduced to solve the VNE problem nowadays. In heuristic VNE algorithms, although some heuristic one-stage VNE algorithms are proposed, they are very complex and not widely used. The two-stage VNE algorithms are easy to implement and they are also widely applied. For example, some previous studies introduce the greedy algorithm into virtual node embedding strategy (e.g. [12]). However, the performance of greedy algorithm is limited. The neighboring nodes in VN might be separated in SN and the corresponding virtual links cost more bandwidth resources. Miao et al. in [27] introduce node/link failure probability into node ranking indicators and propose a self-healing mechanism for reconfigurable service overlay networks.

Above algorithms only take the resource attributes or node/link local attributes into account. It will enlarge the distances between substrate nodes which are neighboring in VN. Topological attribute is gradually introduced into the virtual node embedding. Wang et al. in [13] introduce the network centrality and topology potential into the allocation of virtual resources. These algorithms consider the embedding problem with the topological attribute. Mao et al. in [14] propose a VNE algorithm based on regional resource clustering coefficient. In embedding stage, a node regional resource clustering coefficient evaluation algorithm is proposed, which considers local topology information and resource aggregation extent. This algorithm embeds virtual nodes intensively to the location of abundant resources in SN and decreases embedding costs. Cui et al. in [28] introduce node connection-degree into node attributes. Only the distances between the embedding and the embedded virtual nodes which are adjacent in virtual topology are considered. This method avoids computing all distances between nodes. Li et al. in [29] adopt the top- $k$  dominating model in node ranking stage and propose a novel online embedding algorithm to improve the allocation efficiency of resources.

To make dynamically aware to the network states, some VNE algorithms based on cognitive method are proposed. Cheng et al. in [11] propose a topology-aware VNE. It can sense the topology changes and select the suitable VNE solution. Liang et al. in [18] propose an algorithm of dynamic topology awareness-based reconfigurable service carrying network (RSCN) reconfiguration. The algorithm uses the number of shortest paths as resource critical degree to distinguish substrate resources. And it also dynamically senses the states of critical resources, reoptimises the RSCN according to service request. Gong et al. in [19] propose a trust-aware secure VNE algorithm, in which the concepts of trust relationship and trust degree are introduced into the virtual node resource allocation phase. It also presents a technique for order preference by similarity to an ideal solution (TOPSIS) method which considers the CPU resource, bandwidth resource and centrality indicators. However, the TOPSIS method cannot distinguish the points on perpendicular bisector between the positive and negative ideal solutions. Hu et al. in [20] propose a time and energy aware VNE which takes the running time during the evaluation of substrate nodes and substrate paths into account to improve the performance. Wang et al. in [21] propose a secure VNE scheme to improve the percentage of secure virtual nodes. Zhao et al. in [22] propose a two-stage VNE algorithm based on node adjacent-awareness and path comprehensive evaluation. In the stage of virtual node embedding, it takes resource richness and topology connection feature into account. In the stage of virtual link embedding, a path fitness function is set to find the best path in candidates, which takes the available bandwidth resources, available node resources and hops of path into account.

In this paper, we not only select the proper ranking indicators in VNE, but also research how to integrate them in a new way based on the changing environment.

### 3. Network Model and Evaluation Indicators

The network model and evaluation indications of the VNE problem are formulated and provided in this section.

#### 3.1 Network Model

**SN.** The SN is modeled as a weighted undirected graph  $G_S=(N_S, E_S)$ .  $N_S$  represents the set of substrate nodes.  $E_S$  represents the set of substrate links. In substrate node  $n_s$ ,  $cpu(n_s)$  denotes the available CPU resources and  $loc(n_s)$  denotes the location attribute. In substrate link  $e_s$ ,  $bw(e_s)$  denotes the available bandwidth resources.

**VN.** The VN is modeled as  $G_V=(N_V, E_V)$ .  $N_V$  represents the set of virtual nodes.  $E_V$  represents the set of virtual links. In virtual node  $n_v$ ,  $cpu(n_v)$  denotes the required CPU resources. In virtual link  $e_v$ ,  $bw(e_v)$  denotes the required bandwidth resources.

**VNE.** The VNE is modeled as an embedding function  $M: G_V \rightarrow (N_S', E_S', R_N, R_E)$ .  $(N_S', E_S', R_N, R_E)$  is a subset of  $G_S$ ,  $N_S' \in N_S$  and  $E_S' \in E_S$ .  $R_N$  and  $R_E$  are CPU resources and bandwidth resources assigned to the VN request, respectively. The virtual node embedding and virtual link embedding can be denoted by  $M^N: (N_V, C_N) \rightarrow (N_S', R_N)$  and  $M^E: (E_V, C_E) \rightarrow (E_S', R_E)$ , where  $C_N$  and  $C_E$  refer to the requirements of virtual nodes and links, respectively. An example of VNE problem can be seen in Fig. 1. The results of  $M^N$  and  $M^E$  are  $\{a \rightarrow A, b \rightarrow C, c \rightarrow D\}$  and  $\{(a,b) \rightarrow (A,C), (a,c) \rightarrow (A,D), (b,c) \rightarrow (C,D)\}$ .

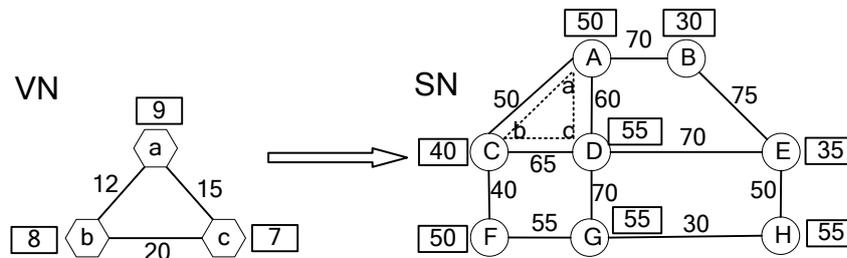


Fig. 1. Example of VNE

#### 3.2 Evaluation Indicators

The VNE problem is a multi-objective optimization problem under the limited resources of the SN. One objective of the VNE algorithm is to maximize the revenue of accepting VN requests. This objective is directly proportional to maximize the number of embedded VN requests. Therefore, the AR of VNE is selected as the main evaluation indicator [11-13]. Another objective is to minimum the costs of accepting VN requests. In different topologies, costs are related to the revenues. It is unfair to compare the costs between different VNE algorithms without considering the revenues. The LTAR/CR is introduced to balance the varying cost values and characterize the performance of VNE in a steady state. Therefore, the LTAR/CR is also selected as the main evaluation indicator [19, 28].

**AR.** It is defined as follows:

$$\chi = \lim_{T \rightarrow \infty} \frac{|VNR_{\text{suc}}|}{|VNR| + \delta_s} \quad (1)$$

Among them,  $\delta_s$  is close to 0.  $|VNR_{\text{suc}}|$  is the number of VN requests that could be embedded onto the SN successfully.  $|VNR|$  is the total number of VN requests.

**Revenue and cost.** In this paper, the universal definitions of revenue and cost are introduced. For VN requests  $G_V=(N_V, E_V)$ , the revenue  $R(G_V, t)$  and the cost  $C(G_V, t)$  are denoted as follows:

$$R(G_V, t) = \alpha \sum_{n_v \in N_V} \text{cpu}(n_v) + \sum_{e_v \in E_V} \text{bw}(e_v) \quad (2)$$

$$C(G_V, t) = \beta \sum_{n_v \in N_V} \text{cpu}(n_v) + \sum_{e_v \in E_V} \text{hops}(e_v) \text{bw}(e_v) \quad (3)$$

Among them,  $\alpha$  and  $\beta$  are weighting coefficients. They are used to balance the CPU and bandwidth resources. They can be adjusted by the infrastructure providers. In this paper, we set  $\alpha=\beta=1$ . In addition,  $\text{hops}(e_v)$  is the hop counts in the substrate link which is embedded by  $e_v$ .

Long-term average revenue  $R(G_S)$  and cost  $C(G_S)$  are used to characterize the performance of VNE in a steady state. They can be defined as follows:

$$R(G_S) = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G_V, t)}{T} \quad (4)$$

$$C(G_S) = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T C(G_V, t)}{T} \quad (5)$$

**LTAR/CR.** It can be used to describe the resource utilization efficiency in the process of VNE, which is denoted as follows:

$$\eta = \frac{R(G_S)}{C(G_S)} = \frac{\lim_{T \rightarrow \infty} \sum_{t=0}^T R(G_V, t)}{\lim_{T \rightarrow \infty} \sum_{t=0}^T C(G_V, t)} \quad (6)$$

#### 4. WRE Method

In two-stage VNE algorithms, most existing virtual node or link embedding strategies are denoted by the fixed formulas. There are three disadvantages about these algorithms. First, it is very difficult to expand or add other ranking indicators if the fixed formula strategies are used. Second, it cannot adapt to the changing environment. Finally, all ranking indicators are not normalized, which will have a significant impact on the fixed formula strategies. In order to solve the above problems, a WRE method to quantify the multiple indicators is proposed.

In WRE method, the relative entropy is introduced to rank the solutions instead of the generalized distance which is commonly used in TOPSIS method. It solves the problem that TOPSIS method is unable to distinguish the point in perpendicular between the positive and negative ideal solutions.

The definition of relative entropy is defined as follows:

$$C = \sum_{i=1}^N [A_i \lg \frac{A_i}{B_i} + (1 - A_i) \lg \frac{1 - A_i}{1 - B_i}] \tag{7}$$

In Eq. (7),  $C$  is the relative entropy of the systems A and B in states  $A_i$  and  $B_i$  ( $i=1,2,\dots,N$ ). The smaller the value is, the smaller the difference of the two system states will be.

The decision matrix  $X$  can be defined as follows:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{pmatrix} \tag{8}$$

In Eq. (8),  $x_{ij}$  is the  $j$ th indicator coefficient of the  $i$ th node. In which,  $i=1,2,\dots,N$ ;  $j=1,2,\dots,M$ .

Owing to the complexity of evaluation relationship in different indicators and the disparity of the indicator coefficient dimensions, the coefficients are normalized. If it is a benefit coefficient, it can be denoted as follows:

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^N x_{ij}} \tag{9}$$

If it is a cost coefficient, it is denoted as follows:

$$r_{ij} = 1 - \frac{x_{ij}}{\sum_{i=1}^N x_{ij}} \tag{10}$$

The normalized decision matrix is  $R = (r_{ij})_{N \times M}$ . In different environments, each indicator has different coefficients. The weighted normalized decision matrix  $Y$  is defined in Eq. (11):

$$Y = R\omega = \begin{pmatrix} r_{11}\omega_1 & r_{12}\omega_2 & \cdots & r_{1M}\omega_M \\ r_{21}\omega_1 & r_{22}\omega_2 & \cdots & r_{2M}\omega_M \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1}\omega_1 & r_{N2}\omega_2 & \cdots & r_{NM}\omega_M \end{pmatrix} \tag{11}$$

In which,  $\omega_j$  is the weighting coefficient of the  $j$ th indicator. Also,  $j=1,2,\dots,M$  and  $\sum \omega_j = 1$ .

Different ideal solutions can be defined as follows:

$$A^+ = \{ \max(y_{i1}, y_{i2}, \dots, y_{iM}) \} = \{ y_1^{\max}, y_2^{\max}, \dots, y_M^{\max} \} \tag{12}$$

$$A^- = \{ \min(y_{i1}, y_{i2}, \dots, y_{iM}) \} = \{ y_1^{\min}, y_2^{\min}, \dots, y_M^{\min} \} \tag{13}$$

$A^+$  and  $A^-$  are the positive and the negative ideal solutions, respectively. The relative entropies of each solution to positive and negative ideal solutions are calculated as follows:

$$C_i^+ = \sum_{j=1}^M \left[ y_j^{\max} \lg \frac{y_j^{\max}}{y_{ij}} + (1 - y_j^{\max}) \lg \frac{1 - y_j^{\max}}{1 - y_{ij}} \right] \tag{14}$$

$$C_i^- = \sum_{j=1}^M \left[ y_j^{\min} \lg \frac{y_j^{\min}}{y_{ij}} + (1 - y_j^{\min}) \lg \frac{1 - y_j^{\min}}{1 - y_{ij}} \right] \tag{15}$$

The similarity between each solution and the ideal one is calculated based on Eq. (16).

$$Z_i = \frac{C_i^-}{C_i^- + C_i^+}, \quad 0 \leq Z_i \leq 1 \quad (16)$$

The details of WRE method are shown in Algorithm 1.

---

**Algorithm 1** The WRE method

---

**Input:** Decision matrix  $X$ , weighting coefficient  $\omega$

**Output:** The importance of node  $i$ ,  $Z_i$

1. Calculate the normalized decision matrix  $R$  (Eq. (9, 10)).
  2. Calculate the weighted normalized decision matrix  $Y$  (Eq. (11)).
  3. The  $A^+$  and  $A^-$  can be calculated based on Eq. (12) and Eq. (13).
  4. The relative entropies  $C^+$  and  $C^-$  can be obtained based on Eq. (14) and Eq. (15).
  5. The similarity of each solution to the ideal one  $Z_i$  can be calculated (Eq. (16)), and it also denotes the node importance. Sort the nodes based on  $Z_i$  from large to small. The higher the similarity is, the higher the importance degree of the nodes in network will be.
- 

As can be seen from the Algorithm 1, there are multiple indicators in normalized decision matrix  $R$  and it is easy to add other indicators in different VNE environments, such as secure VNE and survivable VNE. The weighted normalized decision matrix  $Y$  is constructed by the weighting coefficients and the decision normalized matrix  $R$ . In different VNE environments, we can change the weighting coefficients adaptively to improve the performance of VNE which will be demonstrated in our simulation.

## 5 WRE-CVNE

In this paper, the WRE-CVNE is designed to solve the VNE problem and it can be divided into virtual node embedding and virtual link embedding. In virtual node embedding stage, the WRE method is used to rank the nodes in VN and SN. The node ranking indicators and their weighting coefficients are selected first. Then the WRE method is used to calculate  $Z(n_v)$  and  $Z(n_s)$  in Algorithm 2. In virtual link embedding stage, the WRE method is used to calculate the path importance  $Z(p_i)$  in Algorithm 3.

### 5.1 Virtual Node Embedding

In virtual node embedding, three node ranking indicators are introduced.

**CPU resource.** The CPU resource is used to characterize the node resource indicator. It is shown in Eq. (17).

$$NR(n_i) = CPU(n_i) \quad (17)$$

**Adjacent link bandwidth resource.** It is denoted by the total bandwidth resources of all adjacent links that are connected to the node. It is shown in Eq. (18).

$$DC(n_i) = \sum_{e \in E(n_i)} bw(e) \quad (18)$$

Among them,  $E(n_i)$  is the adjacent link set of node  $n_i$ .

**Node proximity.** Node proximity reflects the importance of node from the topological point of view. It is shown in Eq. (19).

$$CC(n_i) = \frac{1}{\sum_{n_j \in \mathcal{V}(n_i)} d_{ij}} \quad (19)$$

Among them,  $d_{ij}$  is the distance between node  $n_i$  and  $n_j$ . In SN,  $\Psi(n_i)$  is a set of substrate nodes corresponding to the virtual nodes that are adjacent to node  $n_i$  and also have been embedded. The higher the node proximity is, the less substrate link bandwidth resources will be used when virtual link embedded successfully.

In virtual node embedding stage, the CPU resource and adjacent link bandwidth resource are selected as indicators and ranked by WRE method. Then the BFS algorithm is used to rank the ranking results again to take resource and topology attributes into consideration simultaneously. In substrate node ranking stage, the CPU resource, adjacent link bandwidth resource and node proximity are selected to rank the substrate nodes by the WRE method. The BFS algorithm in virtual node ranking is used in conjunction with the node proximity in substrate node ranking. Hence, the adjacent nodes in virtual topology remain close in substrate topology, which greatly reduces the consumption of bandwidth resources.

The steps of virtual node embedding algorithm of the WRE-CVNE are shown in Algorithm 2.

---

**Algorithm 2** Virtual Node Embedding Algorithm of the WRE-CVNE

---

**Input:**  $G_s, G_v$

**Output:** *NodeEmbeddingList*

1. **for** each  $n_v \in N_v$
  2.     **do**
  3.         Select  $NR(n_v)$  and  $DC(n_v)$  as the indicators to calculate  $Z(n_v)$  using the WRE method
  4.     **end for**
  5.     Select virtual node  $n_{vi}$  with the largest  $Z(n_v)$  as the root node to run BFS algorithm and other virtual nodes are divided into several sets according to their distances from  $n_{vi}$
  6.     Rank virtual nodes in each set according to the value of  $Z(n_v)$  from large to small
  7.     Save the results into *VNodeRanking*
  8.     **for** each virtual node in *VNodeRanking*
  9.         Select the candidate substrate node set  $CSNode(n_{vi})$   
Delete the embedded substrate node set  $ESNode$  from  $CSNode(n_{vi})$   
 $CSNode(n_{vi}) = CSNode(n_{vi}) - ESNode$
  10.     **if**  $CSNode(n_{vi})$  is empty
  11.         **Return** NE\_FAILURE
  12.     **else**
  13.         **for** each  $n_s$  in  $CSNode(n_{vi})$
  14.             Select  $NR(n_s)$ ,  $DC(n_s)$  and  $CC(n_s)$  as the indicators to calculate  $Z(n_s)$  using the WRE method
  15.         **end for**
  16.         Embed  $n_v$  to the  $n_s$  which has the largest  $Z(n_s)$  and save it into *NodeEmbeddingList*
  17.         Update the  $ESNode$
  18.     **end if**
  19. **end for**
  20. **Return** NE\_SUCCESS
- 

## 5.2 Virtual Link Embedding

In virtual link embedding stage, the path is denoted as  $p_k = (N_k, P_k)$ , where  $N_k$  is the node set that  $p_k$  crosses and  $P_k$  is the link set that  $p_k$  crosses. Three path ranking indicators are introduced as follows.

**CPU resource balance degree.** It is shown in Eq. (20).

$$NBD(p_k) = \text{ave}_{n_s \in N_k} CPU(n_s) - \min_{n_s \in N_k} CPU(n_s) \quad (20)$$

where  $\text{ave } CPU(n_s)$  and  $\text{min } CPU(n_s)$  are the average and minimum available CPU resources of substrate nodes in  $p_k$ , respectively.

**Bandwidth resource balance degree.** It is denoted as follows:

$$BBD(p_k) = \text{ave}_{e_s \in P_k} bw(e_s) - \min_{e_s \in P_k} bw(e_s) \quad (21)$$

where  $\text{ave } bw(e_s)$  and  $\text{min } bw(e_s)$  are the average and minimum available bandwidth resources of substrate links in  $p_k$ , respectively.

**Path hop counts.**  $\text{hops}(p_k)$  is the hop counts of  $p_k$ . The smaller the  $\text{hops}(p_k)$  is, the less bandwidth resources will be consumed and more remaining bandwidth resources can be used in next VNE.

The details of virtual link embedding algorithm of the WRE-CVNE are given by Algorithm 3.

---

**Algorithm 3** Virtual Link Embedding Algorithm of the WRE-CVNE

---

**Input:**  $G_s, G_v, \text{NodeEmbeddingList}$

**Output:**  $\text{LinkEmbeddingList}$

1. **for** each  $e_v \in E_v$
  2. **do**
  3. Use the  $k$ -shortest path algorithm to find the paths between source substrate node  $u$  and destination substrate node  $q$
  4. Save the  $k$  shortest paths into candidate substrate link set  $\text{CSLink}$
  5. **if**  $\text{CSLink}$  is empty
  6. **Return** LE\_FAILURE
  7. **else**
  8. **for** each  $p_i \in \text{CSLink}$
  9. Select  $\text{NBD}(p_i)$ ,  $\text{BBD}(p_i)$  and  $\text{hops}(p_i)$  as the indicators to calculate the path importance  $Z(p_i)$  using the WRE method
  10. **end for**
  11. Embed  $e_v$  to  $p_i$  with the max  $Z(p_i)$  and save them into  $\text{LinkEmbeddingList}$
  12. **end if**
  13. **end for**
  14. **Return** LE\_SUCCESS
- 

### 5.3 Complexity Analysis

The complexity of the WRE-CVNE includes virtual node embedding complexity and virtual link embedding complexity. In virtual node embedding stage, the complexity of calculating  $Z(n_v)$  of virtual nodes is  $O(|N_v|)$  and  $|N_v|$  is the virtual node number. The complexity of calculating the shortest path between the substrate nodes is  $O(|N_s|^2)$  and  $|N_s|$  is the substrate node number. The complexity of calculating  $Z(n_s)$  of candidate substrate nodes is  $O(|N_v||N_s|^2)$ . In virtual link embedding stage, the complexity of virtual link embedding algorithm of the WRE-CVNE is  $O(k|N_s|(|E_s|+|N_s|\lg|N_s|))$  and  $|E_s|$  is the substrate link number. The total complexity of the WRE-CVNE is  $O(|N_v|+|N_v||N_s|^2+k|N_s|(|E_s|+|N_s|\lg|N_s|))$ .

## 6. Performance Evaluation and Analysis

In this section, we set three experiments to evaluate the WRE-CVNE. At first, we compare the WRE-CVNE with other typical algorithms. Then, the impact of weighting coefficients on virtual node embedding algorithm of the WRE-CVNE is evaluated. Finally, we set three scenarios to evaluate the adaptability of the WRE-CVNE to different environments. In addition, the impact of weighting coefficients in virtual link embedding algorithm of the WRE-CVNE on the performance of algorithm is also verified in different scenarios.

### 6.1 Simulation Environment

In SN, there are 100 substrate nodes and 500 substrate links. The substrate nodes are distributed in  $1000 \times 1000$  distance units. The improved Salam network topology random generate algorithm is introduced to generate the SN and VN topologies. The link connectivity rate is denoted by  $P = n \times e^{\left(\frac{-d^5}{m \times L}\right)}$ , in which,  $m$  and  $n$  are network characteristic parameters as

shown in [22]. Parameter  $m$  controls the relative ratio of short link number to the long ones and parameter  $n$  denotes the whole number of links in network. In this paper,  $m=2.6e^7$ ,  $n=15$  and  $L=1000$ . Parameter  $d$  is the Euclidean distance. The initial available CPU resources of substrate nodes and the bandwidth resources of substrate links are [50, 100], which means they are real numbers following the uniform distribution between 50 and 100.

The arrivals of VN requests follow the Poisson process and the average arrival rate is 0.05. The lifetime of VN requests follows the exponential distribution and the average value is 1000. In VN, the node numbers are [2, 10]. The required CPU and bandwidth resources are [0, 50]. The position constraint value of virtual nodes is  $D=500$ . In the VN, the weighting coefficient of required CPU resource is  $\omega_1$  and the weighting coefficient of required adjacent link bandwidth resource is  $\omega_2$ ,  $\omega_1+\omega_2=1$ ,  $\omega_1/\omega_2=a_1$ . In the SN, the weighting coefficient of CPU resource is  $\omega_3$ , the weighting coefficient of adjacent link bandwidth resource is  $\omega_4$  and the weighting coefficient of node proximity is  $\omega_5$ .  $\omega_3+\omega_4+\omega_5=1$ ,  $\omega_3/\omega_4=a_1$ ,  $(\omega_3+\omega_4)/\omega_5=a_2$ . The weighting coefficient of CPU resource balance degree is  $\omega_6$ , the weighting coefficient of bandwidth resource balance degree is  $\omega_7$  and the weighting coefficient of path hop counts is  $\omega_8$ .  $\omega_6+\omega_7+\omega_8=1$ ,  $\omega_6/\omega_7=a_3$ ,  $(\omega_6+\omega_7)/\omega_8=a_4$ .

In our simulation, the computer is Lenovo Tianyi 510Pro with Windows 10 system. The hardware platform is Inter Core i7-7700 3.6 GHz process with 8 GB RAM. The software platform is Matlab R2007a. All simulations are run for 50000 time units. Each group is run 10 times to take the average values as the final results.

### 6.2 Performance Comparisons of Different VNE Algorithms

The seven VNE algorithms are listed in Table 1 and they use the same SN and VN requests. In the WRE-CVNE,  $a_1=1/9$ ,  $a_2=1/9$ ,  $a_3=1/9$  and  $a_4=1/9$ .

Table 1. Algorithm description

Notation	Description
WRE-CVNE	The algorithm proposed in this paper. In virtual node embedding stage, the virtual nodes are ranked with WRE method by considering the CPU resource, adjacent link bandwidth resource and BFS algorithm. The substrate nodes are ranked with WRE method by considering the CPU resource, adjacent link bandwidth resource and node proximity of nodes. In virtual link embedding stage, the substrate links are ranked with WRE method by considering the CPU resource balance degree, bandwidth resource balance degree and path hop counts.

TC-VNE	The algorithm proposed in [28]. In virtual node embedding stage, the virtual nodes are ranked with the value of $(NR+DC)$ and the substrate nodes are ranked with the value of $((NR+DC)\times CC)$ . In virtual link embedding stage, the $k$ -shortest path algorithm is used.
TC1-VNE	The algorithm proposed in [28]. In virtual node embedding stage, the virtual nodes are ranked with the value of $(NR+DC)$ and the substrate nodes are ranked with the value of $(NR\times DC\times CC)$ . In virtual link embedding stage, the $k$ -shortest path algorithm is used.
BL2-VNE	The algorithm proposed in [12]. The virtual node embedding algorithm is greedy algorithm and the virtual link embedding algorithm is the shortest path algorithm.
TA-SVNE	The algorithm proposed in [19]. In virtual node embedding stage, the virtual nodes and substrate nodes are ranked with TOPSIS method by considering the CPU resource, adjacent bandwidth resource and node centrality. In virtual link embedding stage, the $k$ -shortest path algorithm is used.
CL-VNE	The algorithm proposed in [13]. In virtual node embedding stage, the virtual nodes and substrate nodes are ranked with the node centrality. In virtual link embedding stage, the $k$ -shortest path algorithm is used.
RW-BFS	The algorithm proposed in [11]. It is a one-stage backtracking VNE algorithm based on BFS algorithm, which ranks the virtual nodes and substrate nodes by considering the CPU resource and adjacent bandwidth resource.

As can be seen from Fig. 2-a, the ARs of the BL2-VNE and CL-VNE are close to 0.6. Single VNE indicator is considered and their ARs are the lowest. The RW-BFS is a one-stage VNE algorithm based on BFS algorithm. Its AR is close to 0.67. In the TA-SVNE, three indicators including the CPU resource, adjacent link bandwidth resource and node centrality are taken into account. The performance is greatly improved and the AR is maintained near 0.74. The difference between the TC-VNE and TC1-VNE is the proportion between these three indicators. In this simulation environment, most failures of VNE are due to insufficient bandwidth resources. The TC-VNE increases the proportion of the node proximity. It reduces the hops of substrate paths. Hence, the TC-VNE outperforms the TC1-VNE and shows that same indicators of node ranking with different weighting coefficients have different VNE results. The WRE-CVNE proposed in this paper not only considers the CPU resource, adjacent link bandwidth resource and node proximity, but also the BFS algorithm. The combination of the node proximity in virtual node ranking and the BFS algorithm in substrate node ranking significantly reduces the hop counts of embedded virtual links and saves a large amount of bandwidth resources. Then in virtual link embedding stage, three path ranking indicators are used to reduce the possibility of network fragmentation. Meanwhile, the WRE method is used to increase the weighting coefficients of the node proximity, adjacent link bandwidth resource, bandwidth resource balance degree and path hop counts. Therefore, the performance of the WRE-CVNE is optimal, remained at around 0.81.

In Fig. 2-b, the CL-VNE and BL2-VNE have the lowest ARs and their LTAR/CRs are lower than other five algorithms. The BFS algorithm is used in the RW-BFS which saves bandwidth resources and its LTAR/CR is higher than the CL-VNE and BL2-VNE. The node proximity is used to reduce the hop counts of embedded virtual links in the TC-VNE and TC1-VNE which saves bandwidth resources and reduces the costs. Their LTAR/CRs have increased dramatically. The node centrality is considered in the TA-SVNE to shorten the distance between substrate nodes and increase the LTAR/CR. The WRE-CVNE presented in this paper not only combines the advantages of the TC-VNE and TA-SVNE, but also introduces the BFS algorithm into virtual node ranking stage. The adjacent nodes in virtual topology are embedded priority by topology awareness. The BFS algorithm and node proximity are used in combination to make sure that the adjacent nodes in virtual topology are still close to each other in substrate topology after VNE. The possibility of network

fragmentation is reduced by considering the CPU resource balance degree and bandwidth resource balance degree in virtual link embedding stage. In addition, the weighting coefficients change with the environment. The cost is reduced and the revenue is increased simultaneously.

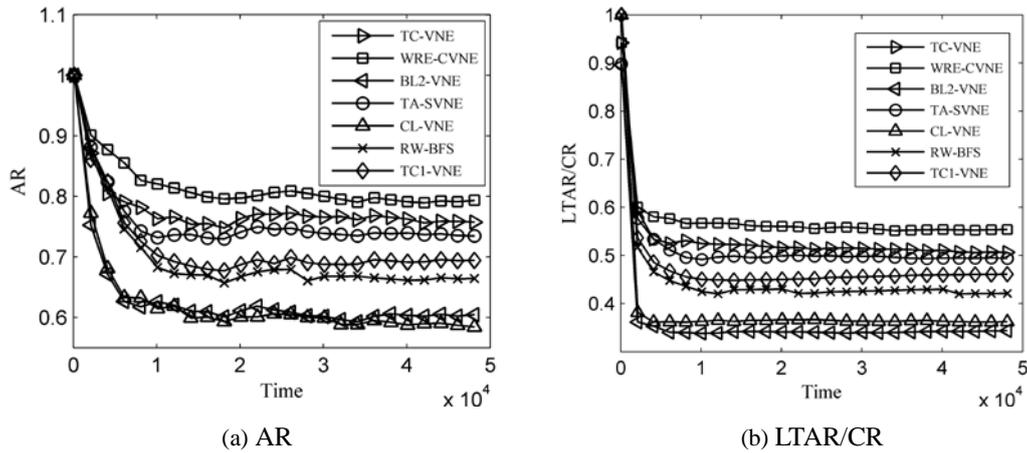


Fig. 2. Comparison between the WRE-CVNE and other VNE algorithms

### 6.3 Performance Comparisons of WRE-CVNE in Different Weighting Coefficients of Virtual Node Embedding Stage

As can be seen from section 6.2, same indicators with different weighting coefficients in virtual node embedding stage have different embedding results. The WRE-CVNE algorithms with different weighting coefficients in virtual node embedding stage are compared in this section. And  $\omega_6=0, \omega_7=0, \omega_8=1$ . The TC-VNE is selected as the comparison algorithm whose performance is optimal except the WRE-CVNE in the previous simulation environment.

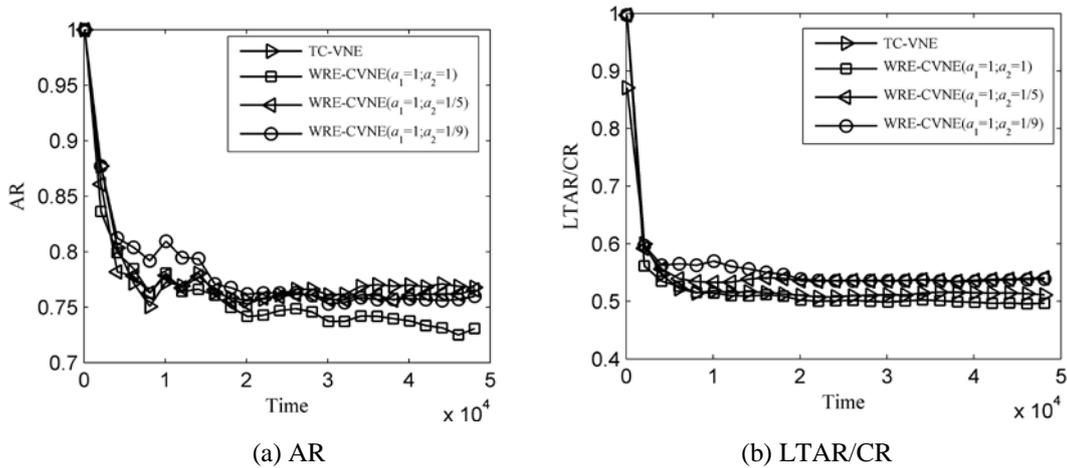
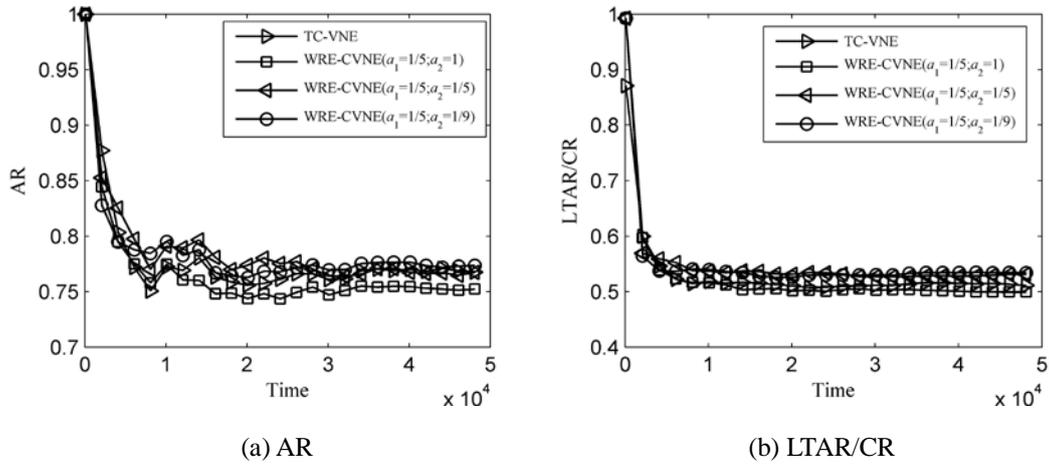


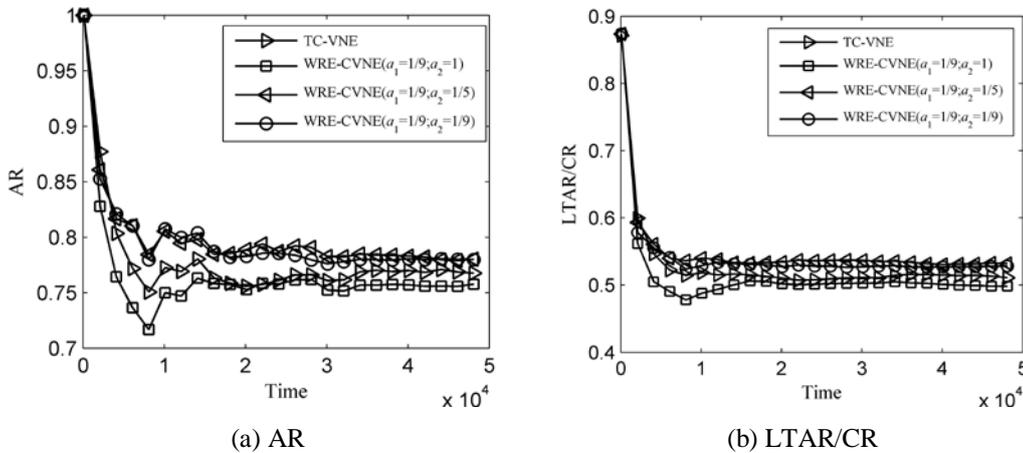
Fig. 3. Comparison of the WRE-CVNE when  $a_1=1$

As can be seen from Fig. 3-a and Fig. 3-b, when  $a_1$  and  $a_2$  are set to 1, the AR of the WRE-CVNE is slightly lower than the TC-VNE. With the decrease of  $a_2$  which means to increase the proportion of the node proximity, when  $a_2$  is set to 1/5 or 1/9, the ARs of the WRE-CVNE and TC-VNE are similar. While the cost is reduced by joint topology awareness in the WRE-CVNE and the LTAR/CR is increased compared with the TC-VNE.



**Fig. 4.** Comparison of the WRE-CVNE when  $a_1=1/5$

As can be seen from **Fig. 4-a**, when  $a_2=1$ , the AR of the WRE-CVNE is lower than the TC-VNE. When  $a_2$  is set to 1/5 or 1/9, the AR of the WRE-CVNE is higher than the TC-VNE. In addition, we can see from **Fig. 4-b** that the LTAR/CR is improved because the hop counts of the substrate links embedded by the WRE-CVNE are reduced.



**Fig. 5.** Comparison of the WRE-CVNE when  $a_1=1/9$

In **Fig. 5-a** and **Fig. 5-b**, when  $a_2$  is 1, the performance of the WRE-CVNE is worse than the TC-VNE. With the increase of the proportion of node proximity, the performance is gradually improved. When  $a_2$  is 1/5 or 1/9, the performance of the WRE-CVNE is better than the TC-VNE.

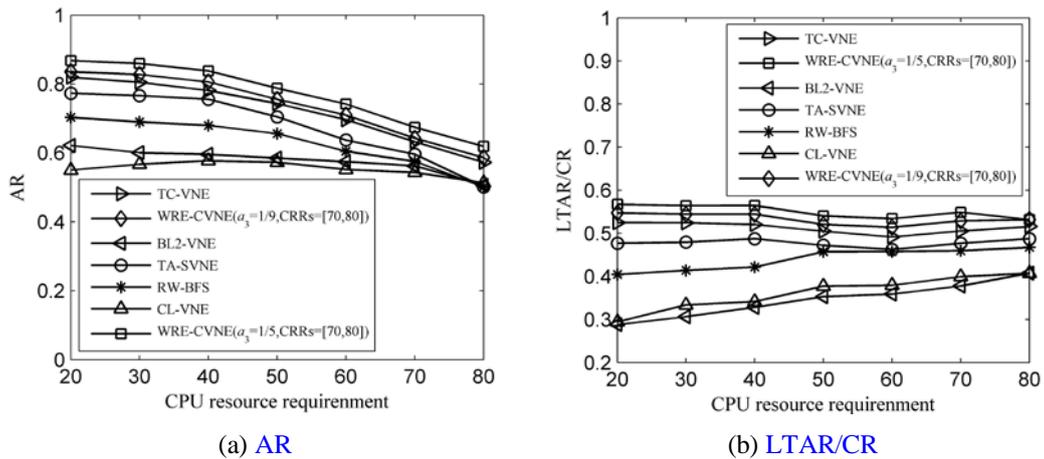
As can be seen from **Fig. 3** to **Fig. 5**, the ARs and LTAR/CRs of WRE-CVNE( $a_1=1/9, a_2=1/5$ ) and WRE-CVNE( $a_1=1/9, a_2=1/9$ ) are obviously higher than the TC-VNE. In the scenario that the bandwidth resources are in short supply and virtual link embedding failure rate is high, increasing the weighting coefficient of the node proximity to reduce the bandwidth resource consumption and increasing the weighting coefficient of the adjacent link bandwidth resource to embed the node whose adjacent bandwidth resources are rich can significantly improve the performance of VNE.

To sum up, the following conclusions can be drawn. First, the weighting coefficient has a great impact on the performance of the WRE-CVNE. In this simulation environment, most failures of VNE are caused by the lack of bandwidth resources. Increasing the weighting coefficients of the adjacent link bandwidth resource and the node proximity can save the bandwidth resources and improve the AR. Second, the setting of weighting coefficient in virtual node embedding stage has a strong universality, for example, when the proportion of the node proximity is greater than 0.5, the performance of the WRE-CVNE is always better than the TC-VNE.

#### 6.4 Performance Comparisons of WRE-CVNE in Different Types of VNs

The performance comparisons of different VNE algorithms in single type of environment are horizontally demonstrated. However, with different demands for CPU resources, bandwidth resources and virtual node number, the types of VNs will be different. In order to demonstrate the performance of the WRE-CVNE in different VN environments, three different VN scenarios are set up in this paper, which correspond to three different VN environments. We also simulate the WRE-CVNE with different weighting coefficients in virtual link embedding stage.

(1) Scenario 1: different CPU resource requirements



**Fig. 6.** Comparison of different VNE algorithms in different CPU resource requirements

In this experiment, the seven algorithms are simulated in different CPU resource requirements. The required bandwidth resources are  $[0, 50]$ . The CPU resource requirements are  $[0, x_1]$ , where  $x_1$  is the upper bound of CPU resource requirements. In WRE-CVNE( $a_3=1/9, CRRs=[70,80]$ ),  $a_3=1/9$  when CPU resource requirements are between 70 and 80. In WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ),  $a_3=1/5$  when CPU resource requirements are between 70 and 80.

As can be seen from **Fig. 6-a** and **Fig. 6-b**, the ARs of all algorithms decrease with the CPU resource requirements increasing. The ARs and the LTAR/CRs of the WRE-CVNE( $a_3=1/9, CRRs=[70,80]$ ) and WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ) are better than other algorithms. In early stage, the bandwidth resources are limited. Increasing the weighting coefficient of the adjacent link bandwidth resource and using the technology of joint topology awareness can shorten the hop counts of embedded virtual links which can save the consumption of bandwidth resources significantly. Comparing the algorithms WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ) and WRE-CVNE( $a_3=1/9, CRRs=[70,80]$ ), we can find

that increasing the weighting coefficient of bandwidth resource balance degree in early stage of virtual link embedding stage can reduce the bandwidth resource fragmentation and save more bandwidth resources to embed next virtual link. With the lack of CPU resources in next stage, increasing the proportion of the CPU resource in virtual node ranking and the CPU resource balance degree in virtual link embedding, the performance of the WRE-CVNE( $a_3=1/5, \text{CRRs}=[70,80]$ ) is always better than other algorithms.

(2) Scenario two: different bandwidth resource requirements

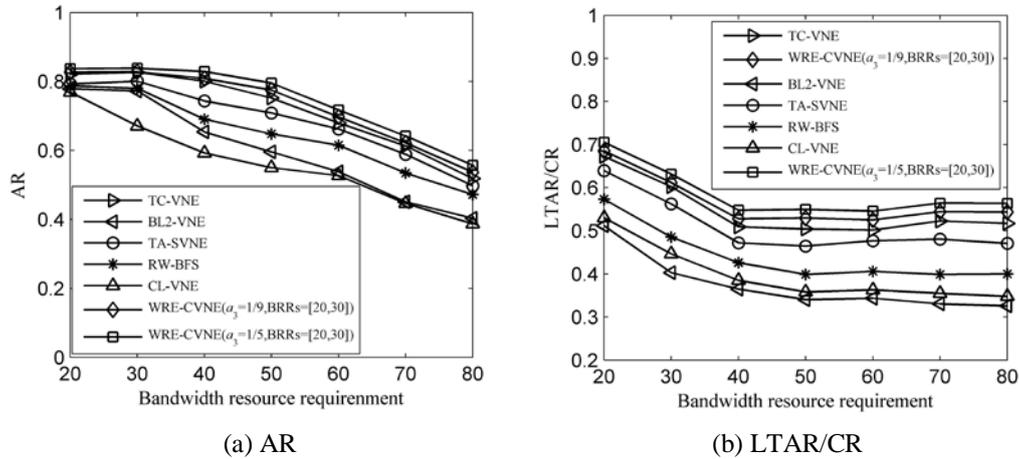


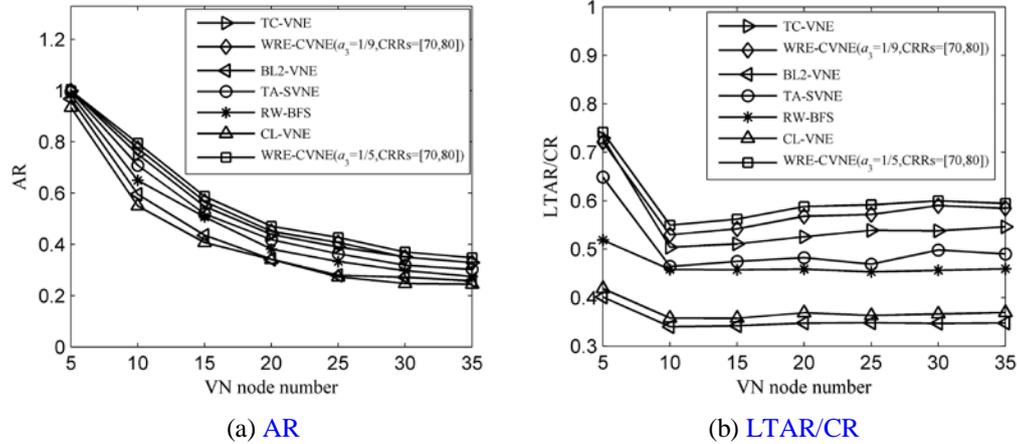
Fig. 7. Comparison of different VNE algorithms in different bandwidth resource requirements

The seven algorithms in different bandwidth resource requirements are simulated in this experiment. The required CPU resources are  $[0, 50]$ . The bandwidth resource requirements are  $[0, x_2]$ , where  $x_2$  is the upper bound of bandwidth resource requirements. In WRE-CVNE( $a_3=1/9, \text{BRRs}=[20,30]$ ),  $a_3=1/9$  when bandwidth resource requirements are between 20 and 30. In WRE-CVNE( $a_3=1/5, \text{BRRs}=[20,30]$ ),  $a_3=1/5$  when bandwidth resource requirements are between 20 and 30.

Fig. 7-a illustrates that with the increase of bandwidth resource requirements, the ARs of all algorithms decrease greatly. The bandwidth resource requirements are relatively small in early stage, and the AR can be improved by increasing the weighting coefficient of the CPU resource. Comparing the algorithms WRE-CVNE( $a_3=1/5, \text{BRRs}=[20,30]$ ) and WRE-CVNE( $a_3=1/9, \text{BRRs}=[20,30]$ ), we can find that increasing the weighting coefficient of the CPU resource balance degree in early stage of virtual link embedding can reduce the CPU resource fragmentation and save more CPU resources to embed next virtual node. Therefore, the AR can be improved significantly especially when CPU resources are limited. With the increase of bandwidth resource requirements, increasing the weighting coefficient of the adjacent link bandwidth resource and bandwidth resource balance degree in virtual link embedding stage can make the superiority of the WRE-CVNE( $a_3=1/5, \text{BRRs}=[20,30]$ ) be more prominent.

Fig. 7-b also illustrates that the performance of the WRE-CVNE( $a_3=1/5, \text{BRRs}=[20,30]$ ) is optimal compared with other algorithms in the area of LTAR/CR.

## (3) Scenario three: different virtual node numbers



**Fig. 8.** Comparison of different VNE algorithms in different virtual node numbers

In this experiment, the seven algorithms are simulated in different virtual node numbers. The required CPU and bandwidth resource are  $[0, 80]$ . The virtual node numbers are  $[2, x_3]$ , where  $x_3$  is the upper bound of virtual node numbers. In WRE-CVNE( $a_3=1/9, CRRs=[70,80]$ ),  $a_3=1/9$  when CPU resource requirements are between 70 and 80. In WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ),  $a_3=1/5$  when CPU resource requirements are between 70 and 80.

As can be seen from **Fig. 8-a**, all algorithms are sensitive to the virtual node number. When the maximum number of virtual nodes is 5, the ARs of all algorithms are maintained at above 0.9. With the increase of virtual node number, the advantages of the WRE-CVNE( $a_3=1/9, CRRs=[70,80]$ ) and the WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ) are becoming more obvious. When  $x$  is close to or greater than 30, the substrate resources have become obviously insufficient relative to VN requirements and all ARs are lower than 0.4. Also, the performance of the WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ) is better than the WRE-CVNE( $a_3=1/9, CRRs=[70,80]$ ), because increasing the weighting coefficient of the CPU resource balance degree can reduce the CPU resource fragmentation when CPU resources are limited. **Fig. 8-b** illustrates that the WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ) is always better than other algorithms in the area of LTAR/CR. It indicates that the WRE-CVNE( $a_3=1/5, CRRs=[70,80]$ ) has the highest resource utilization.

As can be seen from above three experiments, the performance of VNE algorithm will be greatly affected by the resource requirements and virtual node numbers. When CPU resource requirements are large, the AR can be improved by increasing the weighting coefficient of the CPU resource and CPU resource balance degree in the WRE-CVNE. In addition, the technology of joint topology awareness greatly reduces the cost of virtual link embedding. When bandwidth resource requirements are large, the AR and LTAR/CR of the WRE-CVNE are better than other algorithms. When virtual node numbers are changing, the performance of VNE algorithm is greatly affected. The WRE-CVNE has always maintained the optimal AR and LTAR/CR. To sum up, the WRE-CVNE has perfect environmental adaptability by adjusting the weighting coefficients in different VNE environments.

## 7. Conclusions

In this paper, a WRE-CVNE is proposed. It solves the problems of large embedding cost, low AR and environmental adaptability in common VNE algorithms by introducing the cognitive method. First, a WRE method is proposed to quantify the multiple indicators. The introduction of weighting coefficients can make different indicators changing in different VNE environments. It enhances the environmental adaptability and cognitive ability. Then, the BFS algorithm and the node proximity are introduced to realize the combination of virtual topology and substrate topology awareness. Finally, the WRE method is introduced into the virtual link embedding algorithm to reduce the resource fragmentation. We design three experiments to demonstrate the performance of the WRE-CVNE. The first experiment shows the WRE-CVNE has perfect performance in conventional simulation environment. The second experiment verifies the impact of weighting coefficient in virtual node embedding stage on the performance of the WRE-CVNE. The third experiment demonstrates the WRE-CVNE has excellent performance and good adaptability in different scenarios compared with other algorithms by adjusting the weighting coefficients both in virtual node and link embedding stages. The next step is to study the optimal selection of weighting coefficients in the extended WRE-CVNE.

## References

- [1] M. Mashrur, N. Shahriar, R. Ahmed, et al. "Multi-path link embedding for survivability in virtual networks," *IEEE Transactions on Network and Service Management*, vol.13, no.2, pp. 253-266, April, 2016. [Article \(CrossRef Link\)](#)
- [2] M. Raihan and R. Boutaba. "SVNE: Survivable virtual network embedding algorithms for network virtualization," *Network and Service Management*, vol.10, no.2, pp. 105-118, February, 2013. [Article \(CrossRef Link\)](#)
- [3] H. Jiang, Y. Wang, L. Gong, et al. "Availability-aware survivable virtual network embedding in optical datacenter networks," *Journal of Optical Communications and Networking*, vol.7, no.12, pp. 1160-1171, July, 2015. [Article \(CrossRef Link\)](#)
- [4] L. Peterson, T. Anderson, D. Culler, et al. "A blueprint for introducing disruptive technology into the Internet," *ACM SIGCOMM Computer Communication Review*, vol.33, no.1, pp.59-64, July, 2003. [Article \(CrossRef Link\)](#)
- [5] N. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862-876, May, 2010. [Article \(CrossRef Link\)](#)
- [6] A. Bavier, N. Feamster, M. Huang, et al. "In VINI veritas: Realistic and controlled network experimentation," *ACM SIGCOMM Computer Communication Review*, vol.36, no.4, pp. 3-14, October, 2006. [Article \(CrossRef Link\)](#)
- [7] P. Szegedi, S. Figuerola, M. Campanella, et al. "With evolution for revolution: Managing FEDERICA for future Internet research," *IEEE Communications Magazine*, vol.47, no.7, pp. 34-39, July, 2009. [Article \(CrossRef Link\)](#)
- [8] A. Fischer, J. Botero, M. Beck, et al. "Virtual network embedding: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888-1906, February, 2013. [Article \(CrossRef Link\)](#)

- [9] X. Cheng, Z. Zhang, S. Su, et al. "Survey of virtual network embedding problem," *Journal on Communications*, vol.32, no.10, pp. 143-151, October, 2011. [Article \(CrossRef Link\)](#)
- [10] J. Lischka and H. Karl. "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proc. of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, pp. 81-88, August 17-18, 2009. [Article \(CrossRef Link\)](#)
- [11] X. Cheng, S. Su, Z. Zhang, et al. "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol.41, no.2, pp. 39-47, April, 2011. [Article \(CrossRef Link\)](#)
- [12] M. Yu, Y. Yi, J. Rexford, et al. "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 \(CrossRef Link\)](#)
- [13] Z. Wang, Y. Han, T. Lin, et al. "Resource allocation algorithms in the reconfigurable network based on network centrality and topology potential," *Journal on Communications*, vol.33, no.8, pp. 10-20, August, 2012. [Article \(CrossRef Link\)](#)
- [14] Y. Mao, Y. Guo, Z. Wang, et al. "Virtual network embedding algorithm based on regional resource clustering coefficient," *Journal of Electronics & Information Technology*, vol.37, no.10, pp. 2405-2410, October, 2015. [Article \(CrossRef Link\)](#)
- [15] X. Deng, J. Luo, L. He, et al. "Cooperative channel allocation and scheduling in multi-interface wireless mesh networks," *Peer-to-Peer Networking and Applications*, vol.51, no.11, pp. 1-12, November, 2017. [Article \(CrossRef Link\)](#)
- [16] X. Deng, T. He, L. He, et al. "Performance analysis for IEEE 802.11s wireless mesh network in smart grid," *Wireless Personal Communications*, vol.96, no.1, pp. 1537-1555, April, 2017. [Article \(CrossRef Link\)](#)
- [17] X. Deng, Q. Peng, L. He, et al. "Interference-aware QoS routing for neighbourhood area network in smart grid," *IET Communications*, vol.11, no.5, pp. 756-764, May, 2017. [Article \(CrossRef Link\)](#)
- [18] N. Liang, J. Lan, Z. Zhang. "Dynamic topology awareness-based reconfigurable service carrying network reconfiguration," *Journal on Communications*, vol. 37, no. 2, pp. 72-79, February, 2016. [Article \(CrossRef Link\)](#)
- [19] S. Gong, J. Chen, C. Huang, et al. "Trust-aware secure virtual network embedding algorithm," *Journal on Communications*, vol. 36, no. 11, pp. 180-189, November, 2015. [Article \(CrossRef Link\)](#)
- [20] Y. Hu, L. Zhou, H. Chen, et al. "Time and energy aware virtual network embedding using Bayesian theory analysis," *Journal on Communications*, vol.37, no.6, pp. 106-118, June, 2016. [Article \(CrossRef Link\)](#)
- [21] Z. Wang, J. Wu, Z. Guo, et al. "Secure virtual network embedding to mitigate the risk of covert channel attacks," in *Proc. of the IEEE Conference on Computer Communications Workshops*, pp. 144-145, April 10-14, 2016. [Article \(CrossRef Link\)](#)
- [22] Z. Zhao, X. Meng, Y. Su, et al. "Virtual network embedding based on node connectivity awareness and path integration evaluation," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 7, pp. 3393-3412, July, 2017. [Article \(CrossRef Link\)](#)
- [23] L. Wang, H. Qu, J. Zhao, et al. "Virtual network embedding with discrete particle swarm optimization," *Electronics Letters*, vol. 50, no. 4, pp. 285-286, February, 2014. [Article \(CrossRef Link\)](#)

- [24] Q. Zhu, H. Wang, H. Lv, et al. "VNE-AFS: Virtual network embedding based on artificial fish swarm," *Journal on Communications*, vol.33, no.z1, pp. 170-177, December, 2012.  
[Article \(CrossRef Link\)](#)
- [25] Q. Zhu, H. Wang, G. Feng, et al. "VNE-ABC: Virtual network embedding based on artificial bee colony algorithm," *Journal of Beijing University of Technology*, vol. 40, no. 1, pp. 68-73, January, 2014. [Article \(CrossRef Link\)](#)
- [26] L. Chang, H. Gu, Z. Zhang, et al. "Particle swarm optimization user-priority virtual network embedding algorithm," *Journal of Xidian University*, vol. 42, no. 1, pp. 16-22, January, 2015.  
[Article \(CrossRef Link\)](#)
- [27] Y. Miao, C. Wu, Q. Yang, et al. "Self-healing mechanism for reconfigurable service overlay networks," *Journal on Communications*, vol.33, no.8, pp. 52-61, August, 2012.  
[Article \(CrossRef Link\)](#)
- [28] H. Cui, W. Gao, J. Liu, et al. "A virtual network embedding algorithm based on virtual topology connection feature," in *Proc. of the 16th International Symposium on Wireless Personal Multimedia Communications*, pp. 1-5, June 24-27, 2013. [Article \(CrossRef Link\)](#)
- [29] X. Li, H. Wang, B. Ding, et al. "Resource allocation with multi-factor node ranking in data center networks," *Future Generation Computer Systems*, vol. 32, no. 2, pp. 1-12, April, 2014.  
[Article \(CrossRef Link\)](#)



**Yuze Su** is a Ph.D. candidate in Information and Communication Engineering from Air Force Engineering University, China. His research interests include network virtualization and survivable network.



**Xiangru Meng** is a professor at Air Force Engineering University, China. He received his B.S., M.S. and Ph.D. degrees from Xi'an Jiaotong University, China in 1985, 1988 and 1994. From 1995 to 1997, he was a Visiting Scholar at University of Electronic Science and Technology of China. His research interests include next generation Internet, network virtualization and survivable network.



**Zhiyuan Zhao** is a Ph.D. candidate in Computer Application Technology from Air Force Engineering University, China. His research interests include virtual network embedding and software defined network.



**Zhentao Li** is a postgraduate in Computer Science from Air Force Engineering University, China. His research interests include network virtualization and network survivability.