IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 23, 2019, accepted September 8, 2019, date of publication September 13, 2019,
date of current version September 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940971

Virtual Network Embedding Algorithm via Diffusion Wavelet

LEI ZHUANG, SHUAIKUI TIAN, MENGYANG HE, GUOQING WANG, WENTAN LIU, AND LING MA

School of Information and Engineering, Zhengzhou University, Zhengzhou 450001, China

Corresponding author: Lei Zhuang (ielzhuang@zzu.edu.cn)

This work was supported in part by the Key Program of National Natural Science Foundation of China under Grant U1604262, in part by the Key Scientific Research Program of Higher Education of Henan under Grant 19A520003, and in part by the Key Research and Development and Promotion Project in Science and Technology of Henan under Grant 182102210189.

ABSTRACT The great success of the Internet has promoted the development of digital industries and increased the demand for communication bandwidth. For example, ultrahigh-definition videos and vehicle networks require fast bandwidth speed and increase network connection density, respectively. High-bandwidth and high-density parallel communication drive the rapid development of network virtualization and 5G/6G technology. In a network virtualization environment, this new demand also brings new link resource allocation difficulties in existing substrate networks. To solve this far-reaching problem, this paper proposes a virtual network embedding algorithm via diffusion wavelet (VNE_DW), which is an unsupervised structure learning algorithm. Through the diffusion wavelet, the topology structure of nodes, connection density, and link volume among the nodes are comprehensively evaluated. Nodes that facilitate the link mapping success rate are preferentially selected. Experimental results demonstrate that the mapping success rate and revenue-cost ratio of VNE_DW outperform other state-of-the-art algorithms with high bandwidth and density.

INDEX TERMS Virtual network embedding, diffusion wavelet, topology structure, link bandwidth, connection density.

I. INTRODUCTION

Network virtualization is a promising network architecture that can effectively solve network impasse problems [1]. Presently, network virtualization is an integral component of 5G core and data center networks. This architecture is a prerequisite for network slicing, which provides an opportunity for Internet service providers (ISPs) to integrate their devices with standardized high-capacity components [2]. Virtual network embedding (VNE) is an important step in network slice generation, which effectively enhances the flexibility of the network and greatly reduces operating costs while meeting the needs of customized services.

With the advent of the 5G era and the increasing enrichment of cloud services, large video applications have greatly increased the demand for network bandwidth, such as ultrahigh-definition video (e.g., Douyin [3] and YouTube [4]), augmented reality, and virtual reality [5]. These business requirements have led to virtual network requests (VNRs) with different communication intensities and connection densities [6] and introduced new challenges to the deployment of virtual links. The existing VNE algorithm deploys virtual nodes through topology and resource attributes without considering structural characteristics, link volume, and connection density among the nodes. Therefore, the mapping success rate of the link, as well as the overall revenue of the ISP, is reduced especially in the business scenario of high connection density and bandwidth requirements.

Traditional VNE maximizes ISPs' revenue using a two-stage mapping algorithm. After the node mapping ends, the corresponding substrate nodes are connected through a multi-hop link. Therefore, the success rate of the VNE depends on the success rate of the node and link mapping. Selecting an appropriate node position is critical because the mapped node position in the substrate network determines the range of the link mapping. Many researchers are attempting to solve the node mapping scheme in polynomial time using different heuristic algorithms. The metaheuristic algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

is used to solve the VNE problem [7]–[10], and the virtual network mapping scheme is often solved by setting different optimization targets and multiple iterations, which incur a certain time overhead. The heuristic algorithm considers node resource and topology attributes [11]–[13] (e.g., node CPU, node strength, node degree, centrality, etc.) to obtain an approximate optimal mapping scheme through a greedy selection.

Our paper considers structural features, communication capability, and connection density among nodes and focuses on optimizing the scheme of node mapping to facilitate the success rate of link mapping. To analyze structural features, such as connection density, link volume, and closeness among nodes, a virtual network embedding algorithm via diffusion wavelet (VNE_DW) is proposed, which is inspired by spectral theory that treats the diffusion wavelet as the main idea [14], [15].

The main contributions of this paper are as follows.

1. A novel node-ranking algorithm is proposed based on spectral theory, which is extensively used in network presentations and machine learning research spheres. The proposed algorithm is effective in analyzing communication capabilities of a node, communication volume in node pairs, and intimacy among nodes. The algorithm focuses on improving the success rate of link mapping by optimizing the mapping scheme of nodes and obtaining affluent revenue.

2. To make better use of the topology information of nodes, VNE_DW divides the virtual nodes and substrate nodes into different sets according to the mapping state of a node, and determines the mapping scheme of the node by considering the topology relationship between the sets. In addition, this paper considers the multiple topology attributes of the node, and measures the comprehensive effect of these attributes on the nodes through the diffusion wavelet, thus better than evaluating multiple attributes using ordinary mathematical operations. In the link mapping phase, the link mapping success rate is further improved by tailoring links that do not satisfy the bandwidth.

3. Connection density and link volume among nodes are considered in the process of mapping the nodes. The VNE_DW algorithm is better than other traditional algorithms in terms of request acceptance rate and revenue-cost ratio, thereby improving ISPs' total revenue, especially in cases of business requirements with high bandwidth and connection density.

The remainder of this paper is organized as follows. Section 2 presents the current state of research on VNE and graph machine learning algorithms. Section 3 describes and analyzes the VNE problem. Section 4 discusses the VNE_DW algorithm. Section 5 presents the experimental environment and performance evaluation of VNE_DW. Section 6 lists the conclusions.

II. RELATED WORKS

Given that the VNE is an NP-hard problem [16], the algorithms for VNE can be divided into exact-like and heuristic. In the next part, we will divide the algorithms into two parts to detail the research background. In addition, virtual networks have important relationships with graph-related learning algorithms in other fields. Thus, an overview of related research in the field of spectral theory and machine learning is provided.

A. EXACT-LIKE ALGORITHMS

The authors in [17] proposed a VNE algorithm based on mixed integer programming. It is for this reason that the complexity of solving this type of problem is exponential, the authors verified the energy-saving effect of the method on a substrate network with a small number of nodes. In [18], a multidomain VNE algorithm was proposed. The algorithm initially divides the virtual network into multiple subtopologies using the max-flow min-cut algorithm and then maps these sub-topologies in the substrate network through an integer programming in a manner that will minimize the mapping cost. Based on column generation, the authors in [19] proposed a one-shot, unsplittable VNE solution, wherein the problem was modeled as a path-based mathematical program called ''primal.'' The authors created an initial set of paths to solve a dual problem and then solved the primal problem to obtain a final solution. Similar to [19], the authors in [20] presented a method that adopts a compact path-based integer line program, which is solved by leveraging a branch-and-price framework that embeds a column generation process. The idea of column generation is to start with a small set of paths and incrementally incorporate new paths until the optimal path set is contained. Consequently, the complexity of this method exponentially increases with the increase in the number of links.

From the above literature, path-based or mixed integer program solutions are not suitable for solving large-scale or high-link density problems. In addition, the validity of the constructed model determines the effectiveness of the algorithm's solution. Therefore, our work does not consider the virtual network problem of the exact solution but explores the use of heuristic methods in solving the problem.

B. HEURISTIC ALGORITHMS

VNE is a process of finding the logical topology of a virtual network in the substrate network topology, and thus considering topology information will be advantageous in the node mapping phase. Most heuristic algorithms focus on topological attributes to evaluate nodes. The author of [21] proposed a method that simultaneously embeds nodes and links, which is based on subgraph isomorphism detection. Similarly, [22] solved the problem of VNE in a federated cloud environment on the basis of subgraph isomorphism detection. However, the two methods increase the amount of back-trace exploration when resources are insufficient or link connection density is high. Based on the neighborhood method, in which a recursive search is conducted by neighboring nodes on previous nodes, [23] proposed a new general constrained shortest path mapping approach without

considering the node mapping problem. In [24], a node-ranking algorithm that takes into account multiple attributes was proposed. The algorithm utilizes Newton's law of gravity, which combines Euclidean distances among nodes, delay, and resource attributes, to qualitatively evaluate the interaction among nodes. Substrate nodes are sorted using the PageRank algorithm. The virtual node is greedily mapped to the substrate node with the highest ranking to satisfy mapping constraints. The authors of [25] coordinated the advantages of centralized and distributed mapping to formulate a coordinated virtual network mapping algorithm. According to the historical mapping's successful VNRs, the more frequently a substrate node is mapped, the higher the node's priority will be. However, the mapping priority of the virtual node is determined according to the CPU's resource requirement; hence, the neighbor relationship among nodes is ignored. The authors of [26] proposed a virtual network mapping algorithm based on breadth-first search, which ranks nodes according to node degree and clustering coefficient information. When the virtual node is mapped, the substrate node hosting the parent node of the virtual node is determined, and its neighboring nodes are used as candidate nodes. The node that satisfies the constraint in hosting the virtual node is contained in the candidate nodes to reduce link utilization and improve the revenue-cost ratio. The authors of [27] proposed a dynamic virtual network mapping algorithm that solves the problem of allowing the user to dynamically change the virtual network resource requirements and structure after the virtual network is embedded. The algorithm consists of a migration and remapping process. By selecting a node with consideration of the resource utilization but without consideration of the connection state among nodes, it is not conducive to finding a link mapping scheme that satisfies the mapping constraint. The authors of [28] proposed the Monte Carlo tree search algorithm to find the node by considering the connection relationship between the nodes. However, the connection relationship is related to the parent node of the mapping node, and the update of the reward value is determined by the final state, which does not evaluate a better state in the middle.

The above works utilize topologies as heuristic information for node selection while ineffectually incorporating structural features. Moreover, the mapping success rate of the link depends on structural features, such as link connection density, link volume, and closeness among nodes. Link mapping is effectively facilitated when a node that satisfies the resource constraints with a better structure feature is selected.

C. SPECTRAL GRAPH THEORY AND MACHINE LEARNING

With the emergence of graph structure data, such as social, gene regulatory, and brain function networks, the success of deep learning in various applications has prompted researchers to extend machine learning models to non-Euclidean data fields. Consequently, the use of machine learning to analyze geometric topologies is becoming a hot topic. In a supervised learning task, the authors of [29] introduced a new spectral domain convolution structure for deep learning of graphs. The core element of the model is a new class of parametric rational complex functions. Such complex functions can effectively calculate the spectral filter on the graph that specializes in the frequency band of interest, thereby solving the problems of node classification and community detection. Supervised graph structure learning not only requires a large number of manual annotations of data, but the trained model may also have a certain dependence on the data. Therefore, unsupervised structural learning algorithms are urgently needed. The authors of [30] proposed a method for characterizing a node structure as a vector. The similarity of different scales is measured using a hierarchy without nodes or edge attributes and constructing a multilayer graph to encode structural similarity. Then, a structural context of the node is generated. The authors of [31] modified the spectral domain analysis method to analyze the structure through the diffusion wavelet. The nodes in different positions in the graph may have a similar structure. The wavelet coefficient of the node is regarded as the probability, and a characteristic function is used to classify node structures by probability distribution.

This paper utilizes the diffusion wavelet to: (1) learn node structures; (2) analyze connection density, link volume, and closeness among nodes; and (3) select nodes that facilitate link mapping that improves the overall performance of the VNE algorithm.

III. PROBLEM DESCRIPTION AND ANALYSIS

In the network virtualization environment, the infrastructure provider (InP) is responsible for maintaining the substrate network (Fig. 1). The service provider (SP) formulates the corresponding virtual network request based on customer needs and requests the InP to lease resources. The InP benefits from deploying VNRs to the substrate network.

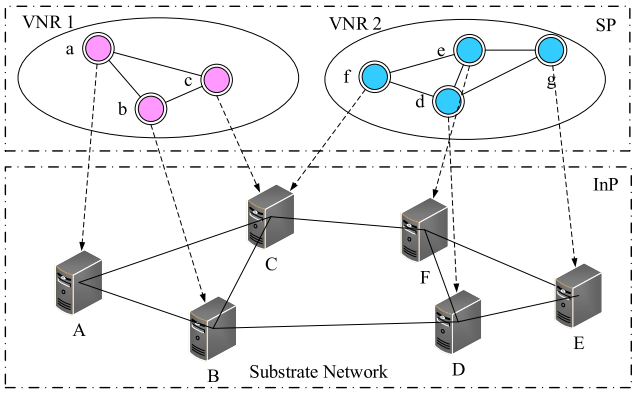


FIGURE 1. Network model.

A. SUBSTRATE NETWORK

The substrate network is represented by an undirected graph $G^s = (V^s, L^s, C^s, B^s)$, where V^s and L^s represent a set of substrate nodes and links, respectively, and C^s and B^s represent the resource capabilities of the substrate nodes and links, respectively.

B. VIRTUAL NETWORK

The virtual network is a topology generated by the network operator according to customer needs. The node and link attributes express a customer's demand for service functions and resources. The virtual network topology is represented by an undirected graph $G^v = (V^v, L^v, C^v, B^v)$, where V^v and L^v represent virtual node and link sets, respectively, and C^v and B^v represent the requirements of virtual node processing and link transmission capabilities, respectively.

C. VNE

The VNE process consists of the node- and link-mapping phases. In the node-mapping phase, candidate nodes that satisfy the virtual node constraint are first identified from the substrate node set. Then, the final substrate node is selected from the candidate nodes using the node-mapping algorithm. In the link-mapping phase, the virtual nodes to which the virtual link is connected are initially determined before selecting the substrate node that hosts the virtual nodes. Then, the substrate link is selected from the substrate nodes that meet the virtual link bandwidth requirement. The virtual node-mapping process follows the scheme (Fig. 1): {a → A, b → B, c → C}, whereas the link-mapping process has the following sequence: {(a, b) → (A, B), (b, c) → (B, C), (a, c) → (A, C)}

D. ANALYSIS OF NETWORK TRAFFIC AND VNE

In two-stage VNE algorithms, considering only the node computing capacity may hinder link mapping due to the low communication capability of the substrate node. Hence, most algorithms adopt the following formula (1) or the improved evaluation to metric the node's communication capability:

$$H(n) = CPU(n) \times \sum_{l \in adjacency(n)} BW(l). \tag{1}$$

The value of the node may be high due to its high computing capability or the adjacent link's high bandwidth capability, especially when the computing and bandwidth requirements are extremely unbalanced. Moreover, the ability to communicate between nodes is not well measured. According to this formula, improper nodes will be selected, resulting in a lower acceptance ratio in the long run. To maximize the acceptance ratio, VNE algorithms do not greedily select the node with the largest metric value but instead select a node that satisfies the computing resource constraint by this priority. However, this discrete node selection method may waste link resources and affect the carrying capacity of the substrate network. For the above factors, we analyze the following five aspects to improve the overall performance of the VNE using structural features of the nodes.

When virtual nodes V_c^v and V_b^v are respectively mapped to substrate nodes V_c^s and V_F^s, the value in the rectangle indicates the computing resource (Fig. 2), whereas the value on the link represents the bandwidth resource. Without loss of generality, we analyze the mapping position of node V_a^v.

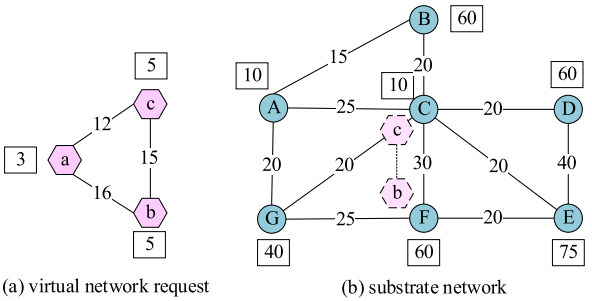


FIGURE 2. Virtual network embedding process.

① When V_a^v is mapped to node V_A^s, link L_{ab}^v is mapped to the path AGF, and link L_{ac}^v is mapped to the path AC. Path AC is a one-hop link that satisfies the L_{ac}^v resource requirement. The mapped path of L_{ab}^v needs to be bridged by node V_G^s, thereby consuming unnecessary link bandwidth and computing resources of V_G^s. This scenario may cause a shortage of resources around intermediate nodes spanned by the substrate path and reduce the carrying capacity of the substrate network to the subsequently arriving virtual network. To shorten the path hop, existing algorithms limit the range of the feasible region for candidate node-mapping, such as hop-limit VNE [32], subgraph isomorphism detection VNE [21], and breadth-first search VNE [26]. However, this process may affect the request acceptance rate to some extent due to the limited node selection.

② When node V_a^v is mapped to node V_B^s, L_{ac}^v is mapped to the path BC, and L_{ab}^v is mapped to the path BAGF. The bandwidth of L_{BA}^s cannot meet the resource constraints of L_{ab}^v due to the link congestion problem, which will cause link-mapping failure. If the link is extremely long, then transmission delay will increase. Therefore, node pairs that are near each other have a lower probability of encountering link congestion than distant ones. Even if the links among the closer pairs of nodes do not meet the resource constraints, they can communicate over longer paths.

③ When node V_a^v is mapped to node V_D^s, L_{ac}^v is mapped to the path DC, and L_{ab}^v is mapped to the path DEF. Although both V_D^s and V_A^s can choose a one-hop link to meet the constraint of L_{ac}^v, the link bandwidth from V_D^s to V_C^s is not larger than that from V_A^s to V_C^s. Therefore, in the node-mapping stage, if V_a^v is mapped to node V_D^s, then the success rate of VNE will be low.

④ When node V_a^v is mapped to node V_E^s, L_{ac}^v is mapped to the path EC, and L_{ab}^v is mapped to the path EF. V_a^v is close to V_c^v and V_b^v with the virtual link to be deployed; therefore, a short path exists between the candidate node V_E^s and node V_C^s, and between node V_E^s and node V_F^s. Selecting a node with such a structural feature will be advantageous for improving the revenue-cost ratio.

⑤ When node V_a^v is mapped to node V_G^s, L_{ac}^v is mapped to the path GC, and L_{ab}^v is mapped to the path GF. V_G^s and V_E^s have the same advantage in terms of revenue-cost ratio. If the bandwidth of L_{GC}^s does not satisfy the resource constraint

of L_{ac}^{v}, two long paths (paths GABC and GAC) will be available between V_G^s and V_C^s. The link-mapping success rate will increase if the connection density of V_G^s to the mapped node V_C^s is high.

In summary, if the greedy choice satisfies node computing resource constraints, then the success rate of the virtual network request-mapping will depend on the link-mapping phase. The success rate of link-mapping depends on the following viewpoints: (1) structural features of the nodes, which evaluate the communication capacity to other nodes; (2) link volume among the nodes; (3) connection density among the nodes; and (4) topological closeness among the nodes. We will use the proposed algorithm to implement these viewpoints.

E. DIFFUSION WAVELETS AND NODE STRUCTURE

The network topology graph $G = (V, E, A, D)$, where $|V| = n$ vertices, $|E| = m$ edges, and A indicates the adjacency matrix of G. If an edge exists between nodes V_i and V_j, then the value of A_{ij} is the weight of the edge, which is defined as follows:

$$Aij = \begin{cases} B(l_{ij}), & \text{if } l_{ij} \in E \\ 0 & \text{, else,} \end{cases} \tag{2}$$

where $B(l_{ij})$ indicates the available bandwidth of link l_{ij}, D indicates the degree matrix of G, and $D_{ii} = \sum_j^n A_{ij}$. In the energy diffusion equation (Eq. 3), L indicates the Laplacian matrix of G, the node is the diffusion source, $\varphi = \{\varphi_1 \cdots \varphi_i\}$ is the intensity distribution of the diffusion sources in G, φ_i is the energy intensity of the node V_i, and the energy transfer between nodes V_i and V_j is set according to Newton's law of cooling [33]. The diffusion coefficient is denoted as K:

$$\frac{d\phi_i}{dt} = -K \sum_j^n A_{ij}(\phi_i - \phi_j). \tag{3}$$

The matrix form of Eq. (3) is as follows:

$$\frac{d\phi}{dt} = -K(D - A)\phi = -KL\phi. \tag{4}$$

A large link bandwidth among nodes signifies more transferred energy per unit time, whereas a high number of neighboring nodes indicates high node energy diffusion per unit time. In short, the diffusion capacity of a node is positively related to connection density, communication capability, and hops of the path among nodes. Subsequently, the above differential equation is solved, and the solution is presented in Eq. (5):

$$\left.\begin{aligned} \phi(t) &= \phi(0) \cdot \Phi \\ \Phi &= U \cdot \begin{bmatrix} e^{-kh_0 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{-kh_n t} \end{bmatrix} \cdot U^{\mathrm{T}} \end{aligned}\right\}, \tag{5}$$

where Φ is the diffusion matrix, $\phi(0)$ represents the energy distribution of nodes of G at time $t = 0$, $\phi(t)$ indicates the energy distribution of nodes at time t, h_i represents the i^{th} eigenvector of L, and Φ reflects the diffusion capacity of the node. Φ_{ij} represents the energy that V_i diffuses toward V_j, and the i^{th} row distribution of Φ reflects the structural feature of the node. The time of energy diffusion is expressed as t. The smaller the value of t is, the more concentrated the energy distribution is (near the node). The structural feature of a node is defined in Eq. (6):

$$S_i = [\Phi_{i1} \cdots \Phi_{in}], \tag{6}$$

where $\Phi_{ii} = 0$ and S_i represent the energy that V_i diffuses to other nodes through multiple paths within time t. A high number of paths and communication capabilities indicates high total energy diffusion. Therefore, the structural metric (SM) of a node can be represented as the total energy diffused to other nodes, which is defined as follows:

$$SM_i = \frac{1}{n-1} \sum_{j \in S_i, j \neq i} S_{ij}, \tag{7}$$

SM_i presents the structural feature of node i. For a more visual analysis of the structure, a well-known topology generation tool (GT-ITM) is used to generate a rand graph with 13 nodes [34]. The SM of the nodes in the graph is visualized by Fig. 3 ($t = 10$ and $K = 1$).

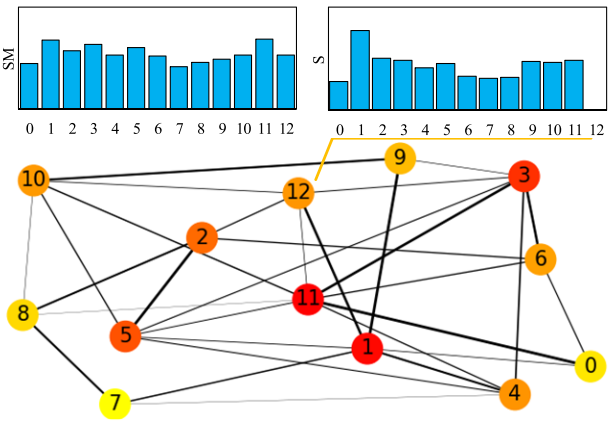


FIGURE 3. Graph SM and S of nodes.

The thickness of the link indicates the currently available bandwidth. The *SM* of a node is indicated by the depth of the node's color. High diffused energy corresponds to high *SM*. A node that is equipped with a high *SM* value is automatically equipped with a good structural feature which metrics communication capacity to other nodes. When the link volume and connection density between node pairs are large, the energy diffusion between node pairs is also large. As indicated by the S distribution of node V_{12} (Fig. 3), the energy value received by node V_9 is greater than that received by node V_8, with the same shortest hops to V_{12}. On the one hand, the larger the node value is, the closer it is to the node V_{12} that acts as the diffusion source (e.g., nodes V_{11} and V_1 are closer to V_{12} than V_9). On the other hand, the topological closeness of V_{10} is relatively close to V_{12},

but because the multiple paths between V_{12} and V_{10} contain links with low volume and communication capability, V_{10} is not as good as other nodes (e.g., V_9). Therefore, the S of V_{10} is relatively low compared to that of V_9.

Thus, wavelet diffusion is a multi-attribute fusion evaluation algorithm rather than a simple multi-attribute for basic mathematical operations (e.g., $+$, $-$, and $\times$). In the next section, the structure of the relationship between a node pair is explored and leveraged to the mapped and candidate nodes.

IV. VNE_DW ALGORITHM DESIGN

The VNE_DW algorithm consists of two phases: the node- and link-mapping algorithms.

A. NODE-EMBEDDING STAGE

A network node often includes multiple resource attributes, such as computing capacity, storage volume, forwarding rate, and so on. Most VNE algorithms evaluate nodes by referencing the amount of computation, determining the priority of candidate nodes according to the proposed node evaluation metric, and following this priority in selecting a node that satisfies resource constraints. However, if the simple summation operation is performed in the multi-attribute comprehensive metric of nodes, ISPs' revenue can be affected in the long term due to the multi-attribute imbalance phenomenon. The proposed algorithm in our paper focuses on optimizing the transmission task, in which node-mapping is directed toward facilitating the success of link-mapping. To adapt to the multi-resource attribute scenario, an evaluation formula for eliminating the imbalance phenomenon is proposed, which is expressed as follows:

$$P(a_r^s, a_r^v) = \left(1 + \exp\left(-\left(a_r^s - a_r^v\right)/a_r^v\right)\right)^{-1}, \tag{8}$$

where a_r^s represents the available resource r of substrate node s and a_r^v represents the resource demand of virtual node v for resource r. The horizontal s axis in Fig. 4 represents the value of a_r^s, whereas the vertical axis is the evaluation value P. The vertical line v is the value of a_r^v. When a_r^s is close to a_r^v, P rapidly increases. However, when a_r^s is extremely large, the function is nearly flat. Therefore, the problem of multi-attribute imbalance caused by a vast single attribute is avoided. The node evaluation formula for a_r^s and a_r^v is as follows:

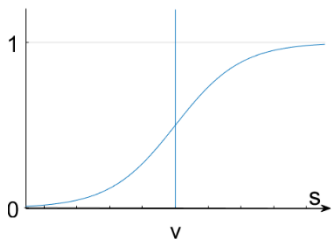


FIGURE 4. Attribute evaluation formula curve graph.

$$NR_s^v = \prod_{r \in RA} \text{sign}(p(a_r^s, a_r^v)) \times p(a_r^s, a_r^v), \tag{9}$$

where "sign" is a symbolic function that returns 1 when the argument is greater than 0.5 and 0 otherwise, and RA is a set of resource attributes.

Eq. (9) is an extension for the multi-attribute node evaluation. In the succeeding discussion, we will determine the mapping order of the virtual nodes and the selection of candidate nodes in a substrate network.

Step 1. Virtual Node Sequence: Different VNE algorithms determine different virtual node embedding sequences according to the specific heuristic information because the embedding order of virtual nodes affects the bearing capacity of the substrate network. For the same reason, our algorithm emphasizes the purpose of improving the success rate of link-embedding by optimizing node selection, along with the consideration of the topological closeness and link volume between the mapping and the mapped node.

First, the first virtual node is determined according to Eq. (10), which has better structural features and higher connectivity:

$$V_i^v = \arg\max_{i \in V^v} (MS_i). \tag{10}$$

Second, the remaining node sequences are determined based on the diffusion distribution of the node V_i^v, as defined in Eq. (11):

$$V_*^v = \arg\text{sort}(S_i), \tag{11}$$

where V_*^v indicates the remaining nodes, which are sorted in descending order according to the diffusion value. According to the diffusion equation, the order of the virtual nodes reflects the node's closeness to V_i^v. The mapping order of the virtual nodes facilitates the increase in the revenue-cost ratio.

Step 2. Selection of the First Substrate Node: As the continuously arriving virtual node is accepted by a substrate node, which gradually degrades the communication capability, the probability of mapping the node decreases. When the consumed substrate node's resources are released, the communication capability of the substrate node improves, thereby increasing the probability of being selected again. According to the above process, the first substrate node hosting the first virtual node is selected according to the following heuristic information.

(1) The proposed algorithm reduces the ISP's cost, which is caused by the reduction in the average mapping path length. Therefore, the distribution of the mapped nodes is concentrated in the local region of the substrate network. For the convenience of the following description, we temporarily refer to the nodes and links of the local region as the mapping region. From the long-term perspective of running the algorithm, the resource distribution status in the mapped region will be consistent. If the resource status in the selected mapping region is good, then the mapping success rate of subsequent nodes will be improved.

(2) A node with strong computing capability may carry a relatively small number of virtual nodes, resulting in large mapping potential. In the mapping region where the node is located, the computing capacity of other nodes may be relatively high due to the consistency of the resource status.

(3) A node with strong communication capability has a relatively high bandwidth of communication paths. In the mapping region, the nodes spanned by the communication paths have strong communication capability in the mapping region.

Considering the above heuristic information, the first candidate node V_i^s is established by the following formula:

$$rank(i) = NR_i^v \times MS_i, \tag{12}$$

where $rank(i)$ indicates the metric evaluation of V_i^s, which has a rich resource and structural feature.

Step 3. Substrate Candidate Nodes Sequence: In the existing node-ranking VNE algorithms, complex constraints in the graph are converted into sequence-mapping relationships, which simplify the difficulty of VNE but lose topological information to some extent. To maximize the use of the topological information, which concentrates the nodes in the local region of the substrate network and facilitates the link-mapping success rate, the virtual node-mapping sequence and the substrate nodes are dynamically divided into four and three sets, respectively. Moreover, the topological relationships among nodes are leveraged to coordinate the relationships among the sets and improve the mapping performance of the algorithm. Fig. 5 illustrates the relations among node sets in the mapping process. The line with an arrow indicates the mapping process or the correspondence relation between the sets of virtual and substrate nodes, whereas the dotted line indicates the topology relationship among the sets.

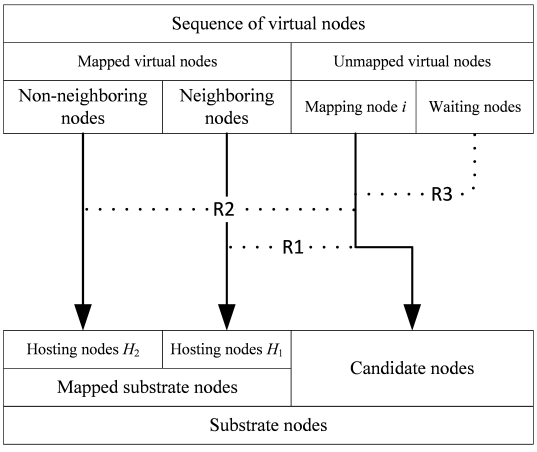


FIGURE 5. Node sets' relationship in the mapping process.

(1) R1 describes the topology relationship between the mapping node and the set of neighboring nodes. The existence of R1 signifies that multiple virtual links must be deployed between a candidate node and the set of hosting nodes H_1. R1 determines the revenue–cost ratio. If the selected candidate node is close to the set of hosting nodes H_1, and the connection density and communication bandwidth of the links are large, then the immediate revenue-cost ratio and the link-mapping rate will increase.

(2) R2 describes the relationship between the mapping node and the set of non-neighboring nodes. Although no link deployment requirement exists between the mapping node and the set of hosting nodes H_2, an R1 relationship may exist between the waiting and non-neighboring nodes. Therefore, maintaining closeness to H_2 can improve the overall aggregation level of nodes and facilitate the improvement of the later revenue-cost ratio.

(3) R3 describes the impact of the mapping node on the set of waiting nodes. When a waiting node becomes the mapping node, R1 and R2 will exist between the waiting nodes and the set of mapped nodes. In this case, a candidate node with better structural features should be selected, which is conducive to communication with subsequent nodes.

Based on the above three items, the metric of the candidate nodes is defined as follows:

$$CN = r_0 \frac{1}{|H1|} \sum_{i \in H1} S_i + r_1 \frac{1}{|H2|} \sum_{i \in H2} S_i + r_2 MS, \tag{13}$$

where H_1 indicates the substrate node set mapped by the neighboring nodes of V_i^v (mapping node i in Fig. 5) and H_2 indicates the set of hosting nodes mapped by the non-neighboring nodes of V_i^v. The summation operation is the superposition of the intensity of wavelet diffusion. r_0, r_1, and r_2 are the weighting factors of R1, R2, and R3, respectively. When r_0 is large, the diffusion of the neighboring nodes has a great influence on the selection of the candidate nodes, and the selected candidate nodes will be close to the neighboring nodes. The metric of the candidate node set CN measures the topological closeness feature and structural features between the candidate nodes and the mapped nodes.

Algorithm 1 Node-Embedding via Diffusion Wavelet

Input: graph G^v, G^s, time t
Output: node-embedding set

1: Sort the virtual nodes in nonincreasing order by using Eq. (11);
2: Map the first virtual node to the substrate node with the higher rank computed using Eq. (12);
3: **For each** vnode **in** unmapped virtual nodes:
4: Flag=FALSE;
5: Sort the substrate nodes in nonincreasing order by using Eq. (13);
6: **For each** snode **in** ordered substrate nodes:
7: Mapping vnode→snode;
8: **If** constraint satisfaction: flag=TRUE; **break**;
9: **If** flag==FALSE: return node mapping fail;
10: **Return** node embedding set;

B. LINK-EMBEDDING STAGE

Link-mapping is divided into single- and multipath mapping, depending on whether the substrate link supports splitting or not. For multipath mapping scenarios, a single set of data flow is circulated over multiple paths. The methods of multipath mapping can reduce mapping costs, reinforce link security, balance the load, etc. Link-mapping between a pair of nodes with high connection density and strong link communication capability will exhibit a high link mapping rate, regardless of whether it is single- or multipath mapping. However, because router support for TCP split transmission is not mature yet, this topic will not be discussed in this paper. For single-path mapping, the shortest path algorithm is used to solve the virtual network link-mapping. By cutting the link that does not meet the bandwidth requirement, the speed in the shortest path algorithm, as well as the success rate of link-mapping, can be increased. The link-mapping algorithm is described as follows.

Algorithm 2 Link-Embedding

Input: graph G^v, G^s
Output: edges embedding set

1: **For each** vedge **in** G^v:
2: Find node*i* and *j* connected by vedge;
3: Find substrate node *si* and *sj* hosting *i* and *j*;
3: Cut the edge of *Gs* below vedge's bandwidth;
4: Find the path between *si* and *sj* by the shortest path algorithm;
5: **If** path = NULL: then return edges embedding fail;
6: **Return** edges embedding set;

C. TIME COMPLEXITY ANALYSIS

The numbers of virtual nodes and links are denoted by $|N^v|$ and $|L^v|$, respectively, whereas the numbers of substrate nodes and links are respectively denoted by $|N^s|$ and $|L^s|$. VNE_DW includes two algorithms. The time complexity of algorithm 1 is $O(|N^v|^3 + |N^s|^3 + |N^s||N^v|)$, whereas the time complexity of algorithm 2 is $O(|L^v|\,|N^s|^2)$.

V. PERFORMANCE EVALUATION

In this section, the settings of the experimental environment are described in detail, and then the evaluation criteria of the algorithm are introduced. Finally, analyses of the experimental results are made in detail.

A. SIMULATION ENVIRONMENT

Similar to [38] and [39], our paper uses GT-ITM [34] to generate topological data required for the experiment. To compare the impact of virtual link factors on the performance of the algorithm, two different substrate environments were set. Environments 1 and 2 explore the impact of link bandwidth and link connection density on algorithm performance, respectively. The results of the experiment are collected after 500 time units.

1) SUBSTRATE NETWORK SETTINGS

This article selects two sizes of networks according to [38], [39]. As shown in Table 1, the resource distribution of nodes and links is uniformly distributed. For the convenience of description, two networks with different scales are named S1 and S2, respectively.

TABLE 1. Substrate network settings.

Substrate network	S1	S2
Number of nodes	50	100
Number of links	130	326
CPU resources	[50,100]	[50,100]
Bandwidth resources	[50,100]	[50,100]

2) VIRTUAL NETWORK SETTINGS

Virtual network settings are shown in Table 2 and 3. The resource distribution of nodes and links follow the uniform distribution. The arriving rate of the VNRs follows the Poisson distribution with an average of five VNRs per 500 time units in substrate network S1, and the arriving rate with an average of five VNRs per 1,000 time units in substrate network S2. The duration of VNRs follow an exponential distribution with an average of 1,000 time units.

TABLE 2. Bandwidth of VNR settings.

Virtual network	Low Bandwidth (LB)	Medium Bandwidth (MB)	High Bandwidth (HB)
Number of nodes	[5,10]	[5,10]	[5,10]
Connection probability of link	0.45	0.45	0.45
CPU resources	[0,20]	[0,20]	[0,20]
Bandwidth resources	[0,20]	[0,30]	[0,40]

TABLE 3. Connection probability of VNR settings.

Virtual network	Low Density (LD)	Medium Density (MD)	High Density (HD)
Number of nodes	[5,10]	[5,10]	[5,10]
Connection probability of link	0.15	0.45	0.75
CPU resources	[0,20]	[0,20]	[0,20]
Bandwidth resources	[0,30]	[0,30]	[0,30]

The remaining parameters are set as follows: $r_0 = 1$, $r_1 = 0.4$, $r_1 = 0.1$, substrate network diffusion time $t = 8$, virtual network diffusion time $t = 3$, and $K = 1$.

3) DESCRIPTIONS FOR THE VALIDATION STEP

To validate the performance of VNE_DW on virtual network requests with different bandwidths and densities, first VNE_DW is validated by using the virtual networks with different bandwidths (Table 2) on the smaller-scale substrate network (S1). Then validate again on the larger-scale substrate network (S2). Second, VNE_DW is validated by using virtual networks with different connection densities (Table 3) on the smaller-scale substrate network (S1) and then validate again on the larger-scale substrate network (S2).

B. VNE EVALUATION CRITERIA

The evaluation criteria used in this paper are represented by Eqs. (14), (15), (16), (17), and (18):

$$AAR = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} VNR_{\text{success}}}{\sum_{t=0}^{T} VNR_{\text{request}}}, \tag{14}$$

where AAR describes the average acceptance rate of the VNE, VNR_{success} indicates the number of successfully embedded VNRs, VNR_{request} indicates the total number of reached VNRs, and T represents the algorithm runtime;

$$RCR = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} R(\mathrm{G^V}, t)}{\sum_{t=0}^{T} C(\mathrm{G^V}, t)}, \tag{15}$$

where RCR indicates the revenue–cost ratio, R represents the revenue of the mapping VNRs, and C represents the cost of mapping VNRs;

$$LFR = \lim_{T\to\infty} \frac{\sum_{t=0}^{T} VNR_{\text{link\_fail}}}{\sum_{t=0}^{T} VNR_{\text{node\_success}}}, \tag{16}$$

where LFR indicates link-mapping failure rate, $VNR_{\text{link\_fail}}$ indicates the number of VNRs with failed link mappings, and $VNR_{\text{node\_success}}$ indicates the number of VNRs with successful node mappings. The higher the LFR is, the higher the failure rate of the link relative to the node mapping is. LFR reveals the impact of link-mapping on the request acceptance rate;

$$LLB = \sqrt{\frac{1}{|L^s|}\sum_{i}^{|L^s|} (\mu\,(li) - AL)^2}, \tag{17}$$

where LLB indicates the link load balancing, μ indicates the link resource utilization, and AL indicates the average link resource utilization. The smaller LLB value reveals the link's ability to provide external services at the lowest cost and best state, so that the link has the highest throughput and higher performance; and

$$AMT = \text{Mean}\left(\sum_{t=0}^{T} MT\,(VNR_{\text{success}})\right), \tag{18}$$

where AMT indicates the average mapping time, MT is the time required for virtual network mapping and Mean is the average value. AMT reveals the resource deployment speed of the VNRs.

C. COMPARATIVE ANALYSIS OF EXPERIMENTAL RESULTS

To assess the performance of our VNE_DW algorithm, we choose three state-of-the-art algorithms for comparison: (1) the Markov Chains with Rewards Ranking (MCRR) method [35], which uses Markov random walk as the model and computes the priority of candidate nodes through the resources of neighbor nodes; (2) the Virtual Network Embedding based on Elimination and Choice Expressing Reality (ELECTRE_VNE) method [36], which is a multi-attribute evaluation method that combines topological and resource attributes to evaluate the priority of nodes; and (3) the Energy Efficient, Concurrent and Topology-Aware virtual network embedding (EE_CTA) method [37], which focuses on network topology by assigning reachability rank to resources, and adopts a genetic algorithm to solve the VNE scheme.

As shown in Fig. 6, with the distribution of link bandwidth and density requested by the virtual network increasing, the average resource requirement of each virtual network also increases, and thus the link failure rate (LFR) of the five algorithms increases. However, as shown in subpictures a and b, the LFR of VNE_DW is lower than that of the other three algorithms as the bandwidth increases. Especially in HB, VNE_DW is 20.2% and 22.2% lower than the average value of the other three algorithms. Similarly, as shown in subpictures c and d, when the density is HD, the LFR of VNE_DW is 14.6% and 19.2% lower than the average

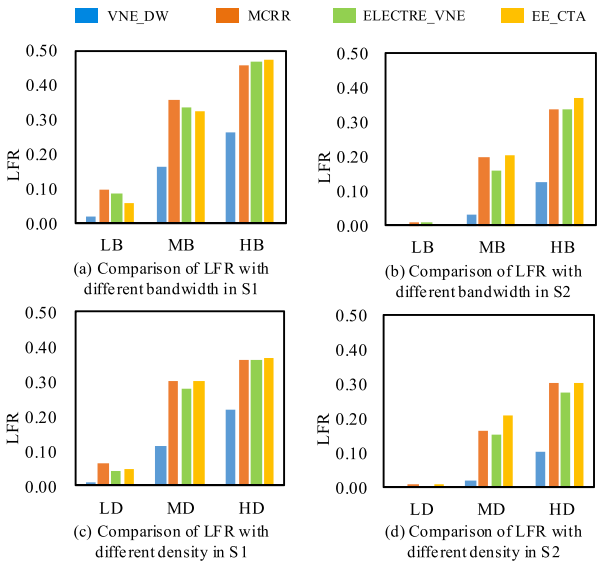


FIGURE 6. LFR over bandwidth and density.

value of the other three algorithms. In terms of different bandwidths and densities, VNE_DW has the lowest LFR among the four. The main reason for this phenomenon is that, when selecting a node that satisfies the resource constraint, the VNE_DW algorithm preferentially selects a node that exhibits a neighborhood relationship with the mapped node (i.e., in terms of path bandwidth and connection density among the nodes). Furthermore, VNE_DW considers the impact of the current mapping node on the remaining nodes and then selects a node with strong communication capability to improve the success rate of link-mapping among subsequent nodes. When all nodes are mapped, the LFR will decrease. MCRR takes advantage of the topological attribute to calculate the resource of the node. However, node selection does not consider path bandwidth and the relationship among nodes; hence, the LFR increases. ELECTRE_VNE considers the node's local resources and closeness among the nodes and uses the shortest hops to evaluate the nodes. However, this evaluation only reflects the closeness among nodes; the path bandwidth between the candidate and the mapped nodes and the link connection relationship of the virtual node are not accounted for. Therefore, when the selected candidate node is close to a mapped node, the virtual link is deployed between the candidate and other mapped nodes with multiple hops, thereby increasing the link-mapping failure rate. EE_CTA uses reachability graphs to mark the reachability of nodes and links. Nodes with the same reachability belong to the same subgraph and are close to each other. EE_CTA improves the degree of aggregation between mapped nodes by minimum reachability. However, the distribution of nodes is concentrated and their communication capabilities may not be very good, which is not conducive to reducing LFR.

As observed from Fig. 7, the revenue-cost ratio (RCR) decreases with the increase in link bandwidth and connection density of VNRs, because of the increasing mapping load of the link. As shown in subpictures a and b, when the distribution of bandwidth is HB, VNE_DW is 22.4% and 27.6% higher than the average value of the other three algorithms. As shown in subpictures c and d, when the density of the link is HD, VNE_DW is 18.7% and 23.1% higher than the average value of the other three algorithms. The reason for this phenomenon is because the VNE_DW algorithm preferentially maps the nodes close to neighbors and reduces the hop count of the substrate path. Moreover, the mapping position is close to the mapped nodes, which aggregates the distribution of the mapped nodes, consequently improving the RCR. The MCRR algorithm selects a resource-rich node as a candidate node. When the virtual and candidate nodes are sorted, the number of path hops increase because the connection relationship among the nodes is not considered, thereby reducing the RCR. The ELECTRE_VNE algorithm considers the path hops between the candidate node and the mapped node set. However, each selected substrate node is close to a node of the mapped set. The virtual link will be deployed not between the node pairs, but between other node pairs. When more attributes are considered, the impact of the hop attribute on the ranking may decline, resulting in a low RCR. EE_CTA is a multi-objective optimization algorithm, and multiple targets of VNE tend to compete with each other. However, EE_CTA does not combine topology information well to guide nodes to adjust the mapping scheme. Instead, the nodes are randomly adjusted. The final solution obtained by the nondominated sorting algorithm is not necessarily the highest RCR.

The decline in average acceptance rate (AAR) as link bandwidth and density increasing are depicted in Fig. 8. Because the total resource of S1 is less than that of S2, the AAR on S1 falls faster than the AAR on S2. As shown in subpictures a and b, when the distribution of bandwidth is HB, VNE_DW is 18.1% and 18.3% higher than the average value of the other three algorithms. However, in subpictures c

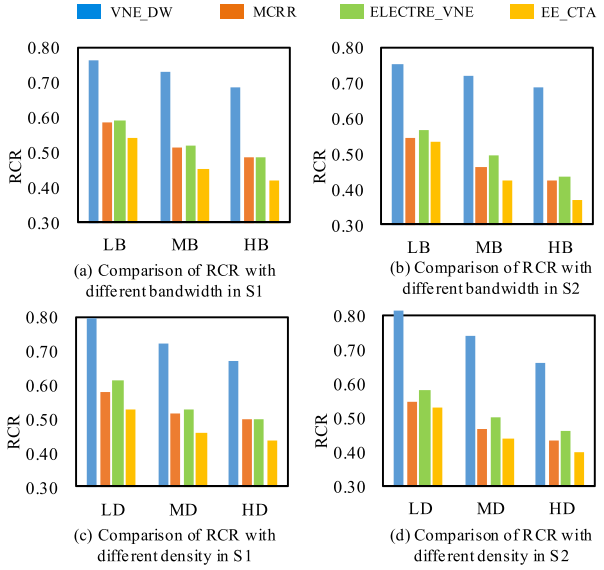


FIGURE 7. RCR over bandwidth and density.

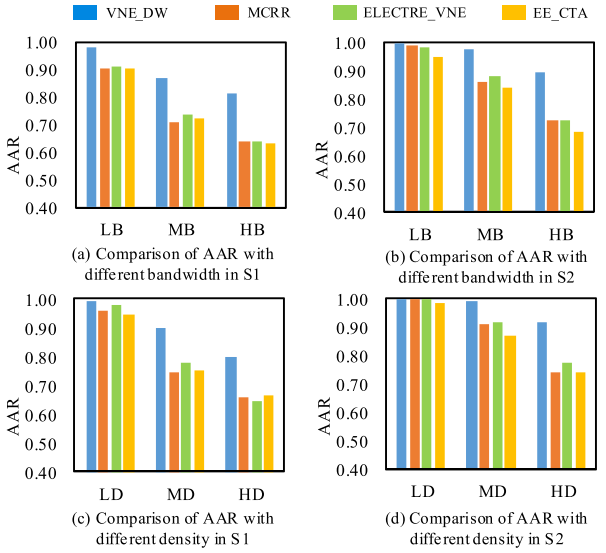


FIGURE 8. AAR over bandwidth and density.

and d, when the density of the link is HD, VNE_DW is 14.1% and 16.6% higher than the average value of the other three algorithms. For limited substrate resources, as the connection density and bandwidth of the virtual link increase, the consumption of the substrate resources continuously increases. When the substrate resources cannot carry the VNR, the AAR of all algorithms shows a downward trend. We consider the link constraint between direct neighbors, indirect neighbors, and subsequent unmapped nodes to reduce unnecessary bandwidth consumption and improve the accommodation of the substrate network. Unnecessary bandwidth resources are consumed when the nodes are mapped using MCRR, ELECTRE_VNE, and EE_CTA. Moreover, MCRR, ELECTRE_VNE, and EE_CTA have a higher LFR, thereby resulting in a lower acceptance rate compared to VNE_DW. The EE_CTA algorithm generates a new virtual network mapping scheme through crossover and mutation operations, in which the crossover operation is prone to infeasible solutions, such that individuals in the population are eliminated during the evolution process, thereby reducing the population richness. which affects the AAR of the VNE in the long run.

Fig. 9 shows that link load balancing (LLB) presents different tendencies with different bandwidths and densities over network sizes. In the case of lower bandwidth and lower density, the LLB of VNE_DW is poor. As shown in subpictures a and b, when the distribution of bandwidth is LB, VNE_DW is 0.049 and 0.078 higher than the average value of the other three algorithms. However, when the distribution of bandwidth is HB, VNE_DW is 0.047 and 0.045 lower than the average value of the other three algorithms. As shown in subpictures c and d, when the density of the link is LD, VNE_DW is 0.039 and 0.065 higher than the average value of the other three algorithms. However, when the density of link is HD, VNE_DW is 0.023 and 0.041 lower than the average value of the other three algorithms. In the lower density or bandwidth environment, the VNE_DW has higher LLB than

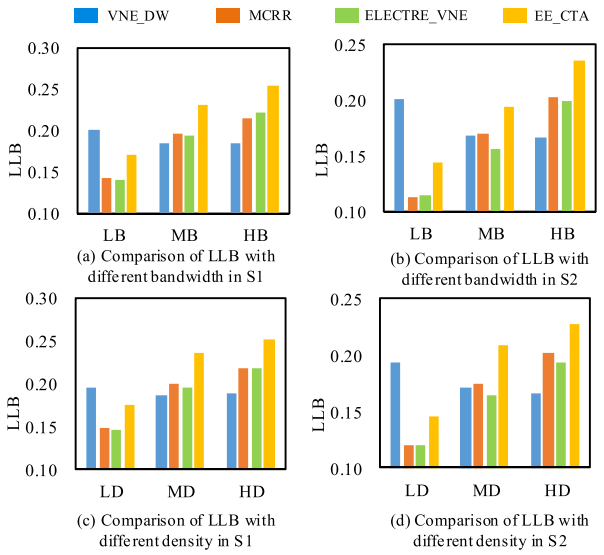


FIGURE 9. LLB over bandwidth and density.

the other algorithms, but with the increase of bandwidth and density, the LLB begins to decrease, the balance performance improves, and the LLB of the other algorithms begins to increase. The reason is that the substrate resources are relatively sufficient, and the impact on the link is not enough to change the mapping scheme of the nodes. Therefore, balance performance is relatively poor. However, as the bandwidth and density increase, VNE_DW actively searches for a node with link resources sufficient for mapping, reducing the load on low- resource links. Other algorithms, when selecting nodes, do not consider the communication capabilities between nodes well, so the link bearer is unbalanced after greedy selection of nodes.

In summary, VNE_DW exhibits higher AAR and RCR compared with other algorithms, thereby improving the overall revenue of network operators.

Table 4 shows the average mapping time (AMT) of the five algorithms. VNE_DW has a slightly higher AMT than MCRR, but lower mapping time than ELECTRE_VNE and EE_CTA. VNE_DW needs to solve the eigenvalues and eigenvectors of the Laplacian matrix, based on the wavelet diffusion coefficient matrix. The node structure is analyzed, and the candidate order is determined. The MCRR needs to obtain the smooth distribution of the nodes using the Markov matrix and determine the mapping order by sorting the candidate nodes. The ELECTRE_VNE algorithm needs to calculate multiple node attributes, and the hop count attribute of the substrate network (to the mapping node) is necessary, when selecting a candidate node for each virtual node. The attribute of each candidate solution must be compared and analyzed, which is a time-consuming process. EE_CTA requires multiple iterations of the population, and a large number of mapping schemes need to be evolved or eliminated, so it takes a long time.

TABLE 4. Average mapping time.

Algorithms	AMT
VNE_DW	0.0187 seconds
MCRR	0.0156 seconds
ELECTRE_VNE	0.4849 seconds
EE_CTA	3.7001 seconds

VI. CONCLUSION

This paper preferentially selected nodes with better structural features to coordinate link-mapping in two-stage VNE algorithms. Wavelet diffusion, which is an algorithm for unsupervised learning node structure, is utilized. The results show that the diffusion process can measure link bandwidth and connection density among nodes. This method is suitable for the needs of communication traffic with intensive services in environments with poor communication links.

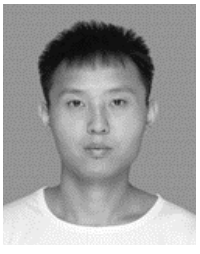
REFERENCES

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2015.

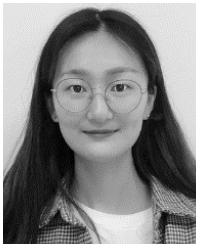
[2] K. Hejja and X. Hesselbach, "Online power aware coordinated virtual network embedding with 5G delay constraint," *J. Netw. Comput. Appl.*, vol. 124, pp. 121–136, Oct. 2018.
[3] *Douyin*. Accessed: Apr. 2019. [Online]. Available: https://www.douyin.com
[4] *Youtube*. Accessed: Apr. 2019. [Online]. Available: https://www.youtube.com
[5] T. Chris, "Deep inside the Internet backbone: How content delivery networks deal with bandwidth demands from OTT services, 4K, virtual reality—And viewers," *Broadcast. Cable*, vol. 147, no. 16, pp. 8–11, Jul. 2017.
[6] F. Dong, X. Guo, P. Zhou, and D. Shen, "Task-aware flow scheduling with heterogeneous utility characteristics for data center networks," *Tsinghua Sci. Technol.*, vol. 24, no. 4, pp. 400–411, Aug. 2019.
[7] P. Zhang, H. Yao, C. Fang, and Y. Liu, "Multi-objective enhanced particle swarm optimization in virtual network embedding," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 167–175, Dec. 2016.
[8] H. Zheng, J.-J. Li, Y.-J. Gong, W.-N. Chen, Z. Yu, Z.-H. Zhan, and Y. Lin, "Link mapping-oriented ant colony system for virtual network embedding," in *Proc. IEEE Conf. Evol. Comput. (CEC)*, San Sebastian, Spain, Jun. 2017, pp. 1223–1230.
[9] M. He, L. Zhuang, S. Tian, G. Wang, and K. Zhang, "Multi-objective virtual network embedding algorithm based on Q-learning and curiosity-driven," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 150–161, Jun. 2018.
[10] Y. Yuan, C. S. Wang Peng, and S. Keshav, "Topology-oriented virtual network embedding approach for data centers," *IEEE Access*, vol. 7, pp. 2429–2438, Dec. 2018.
[11] X. Cheng, S. Su, Z. Zhang, K. Shuang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology awareness and optimization," *Comput. Netw.*, vol. 56, no. 6, pp. 1797–1813, Apr. 2012.
[12] Z. Wang, Y. Han, T. Lin, Y. Xu, S. Ci, and H. Tang, "Topology-aware virtual network embedding based on closeness centrality," *Frontiers Comput. Sci.*, vol. 7, no. 3, pp. 446–457, Jun. 2013.
[13] J. Liao, J. Liao, J. Wang, S. Qing, and Q. Qi, "Topology-aware virtual network embedding based on multiple characteristics," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 1, pp. 145–164, Jun. 2014.
[14] O. K. Johnson, J. M. Lund, and T. R. Critchfield, "Spectral graph theory for characterization and homogenization of grain boundary networks," *Acta Mater.*, vol. 146, pp. 42–54, Mar. 2018.
[15] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 53–94, 2006.
[16] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electron. Notes Discrete Math.*, vol. 52, pp. 213–220, 2016.
[17] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. de Meer, "Energy efficient virtual network embedding," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 756–759, May 2012.
[18] I. Houidi, W. Louati, W. B. Ameur, and D. Zeghlache, "Virtual network provisioning across multiple substrate networks," *Comput. Netw.*, vol. 55, no. 4, pp. 1011–1023, 2011.
[19] R. Mijumbi, J. Serrat, J.-L. Gorricho, and R. Boutaba, "A path generation approach to embedding of virtual networks," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 3, pp. 334–348, Sep. 2015.
[20] Y. Wang, Q. Hu, and X. Cao, "A branch-and-price framework for optimal virtual network embedding," *Comput. Netw.*, vol. 94, pp. 318–326, Jan. 2016.
[21] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proc. ACM Workshop Virtualized Infrastruct. Syst. Archit.*, Barcelona, Spain, 2009, pp. 81–88.
[22] A. Aral and T. Ovatman, "Network-aware embedding of virtual machine clusters onto federated cloud infrastructure," *J. Syst. Softw.*, vol. 120, pp. 89–104, Oct. 2016.
[23] D. Chemodanov, P. Calyam, F. Esposito, and A. Sukhov, "A general constrained shortest path approach for virtual path embedding," in *Proc. IEEE. LANMAN*, Rome, Italy, Jun. 2016, pp. 1–7.
[24] H. Cao, L. Yang, and H. Zhu, "Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 108–120, Feb. 2018.
[25] M. Feng, J. Liao, S. Qing, T. Li, and J. Wang, "COVE: Co-operative virtual network embedding for network virtualization," *J. Netw. Syst. Manage.*, vol. 26, no. 1, pp. 79–107, Mar. 2017.
[26] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on the degree and clustering coefficient information," *IEEE Access*, vol. 4, pp. 8572–8580, 2016.
[27] K. D. Chinmaya and K. S. Prasan, "DYVINE: Fitness-based dynamic virtual network embedding in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1029–1045, May 2019.
[28] S. Haeri and L. Trajković, "Virtual network embedding via monte carlo tree search," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 510–521, Feb. 2018.
[29] R. Levie, M. Federico, X. Bresson, and B. Xavier, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, Jan. 2019.
[30] F. R. Ribeiro, H. P. Savarese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. 23th ACM SIGKDD Int. Conf.*, Halifax, NS, Canada, Jul. 2017, pp. 385–394.
[31] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proc. 24th ACM SIGKDD Int. Conf.*, London, U.K., Jun. 2018, pp. 1320–1329.
[32] M. Zhang and X. Tang, "Hop-limit path mapping algorithm for virtual network embedding," *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2033–2048, Aug. 2017.
[33] M. I. Davidzon, "Newton's law of cooling and its interpretation," *Int. J. Heat Mass Transf.*, vol. 55, no. 21, pp. 5397–5402, Oct. 2012.
[34] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Mar. 1996, pp. 594–602.
[35] F. Bianchi and F. L. Presti, "A Markov reward model based greedy heuristic for the virtual network embedding problem," in *Proc. IEEE MASCOTS*, London, U.K., Sep. 2016, pp. 373–378.
[36] P. Zhang, H. Yao, C. Qiu, and Y. Liu, "Virtual network embedding using node multiple metrics based on simplified ELECTRE method," *IEEE Access*, vol. 6, pp. 37314–37327, Jun. 2018.
[37] A. Jahani, L. M. Khanli, M. T. Hagh, and M. A. Badamchizadeh, "EE-CTA: Energy efficient, concurrent and topology-aware virtual network embedding as a multi-objective optimization problem," *Comput. Standards Interfaces*, vol. 66, Oct. 2019, pp. 103351–103368.
[38] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.
[39] X. Chang, X. M. Mi, and J. K. Muppala, "Performance evaluation of artificial intelligence algorithms for virtual network embedding," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2540–2550, Nov. 2013.

LEI ZHUANG was born in Rizhao, Shandong, China, in 1963. She received the B.S. degree in computer science from Zhengzhou University, Zhengzhou, China, in 1986, the M.S. degree in computer software from Huanghe University, Zhengzhou, in 1988, and the Ph.D. degree in computer software and theory from the PLA Information Engineering University, Zhengzhou, in 2004.

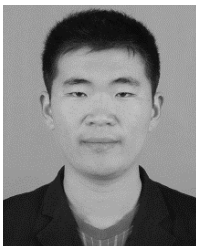
She is currently a Professor and a Ph.D. Supervisor with Zhengzhou University. Her main research interests include future network architecture, network virtualization, and model checking.



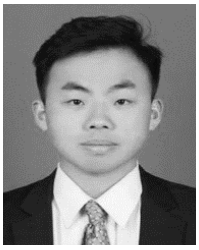
SHUAIKUI TIAN was born in Shangqiu, Henan, China, in 1992. He received the B.S. degree in electronic and information engineering from the Anyang Institute of Technology, Anyang, China, in 2017. He is currently pursuing the M.S. degree in computer science and technology with Zhengzhou University. His main research interests include next generation internet, network virtualization, and machine learning.



MENGYANG HE was born in Nanyang, Henan, China, in 1994. She received the B.S. degree in embedding systems from Zhengzhou University, Zhengzhou, China, in 2015, where she is currently pursuing the Ph.D. degree in software engineering. She has been taking successive postgraduate and doctoral programs of study for doctoral degree, since 2015. Her main research interests include next generation internet, network virtualization, and machine learning.



GUOQING WANG was born in Linyi, Shandong, China, in 1989. He received the B.S. degree in computer science and technology and the dual degrees in business administration from Zhengzhou University, Zhengzhou, China, in 2011, where he is currently pursuing the Ph.D. degree in software engineering. He has been taking successive postgraduate and doctoral programs of study for doctoral degree, since 2013. His main research interests include model checking, formal analysis, and the IoT security.



WENTAN LIU was born in Shangqiu, Henan, China, in 1995. He received the B.S. degree in software engineering from Zhengzhou University, Zhengzhou, China, in 2017, where he is currently pursuing the M.S. degree in software engineering. His main research interests include next generation internet, network virtualization, and machine learning.

LING MA was born in Pingdingshan, Henan, China, in 1963. He received the B.S. degree in mathematics from Zhengzhou University, Zhengzhou, China, in 1984, the M.S. degree in applied mathematics from the Hebei University of Technology, Tianjin, China, in 1987, and the Ph.D. degree in manufacturing engineering from Beihang University, Beijing, China, in 1997.

He is currently a Professor and a Ph.D. Supervisor with Zhengzhou University. His main research interests include pattern recognition, image processing, and machine vision.