

IoV Scenario: Implementation of a Bandwidth Aware Algorithm in Wireless Network Communication Mode

Peiyong Zhang , Chao Wang, Gagangeet Singh Aujla , *Senior Member, IEEE*,
Neeraj Kumar , *Senior Member, IEEE*, and Mohsen Guizani , *Fellow, IEEE*

Abstract—Wireless network communication has developed rapidly in recent years, especially in the field of Internet of vehicles (IoV). However, due to the limitations of traditional network architecture, resource scheduling in wireless network environment is still facing great challenges. We focus on the urgent need of users for bandwidth resources in the IoV scenario under virtual network environment. This paper proposes a bandwidth aware multi domain virtual network embedding (BA-VNE) algorithm. The algorithm is mainly aimed at the problem that users need a lot of bandwidth in wireless communication mode, and solves the problem of bandwidth resource allocation from the perspective of virtual network embedding (VNE). In order to improve the performance of the algorithm, we introduce particle swarm optimization (PSO) algorithm to optimize the performance of the algorithm. In order to verify the effectiveness of the algorithm, we have carried out simulation experiments from link bandwidth, mapping cost and virtual network request (VNR) acceptance rate. The final results show that the proposed algorithm is better than other representative algorithms in the above indicators.

Index Terms—BA-VNE Algorithm, bandwidth requirement, virtual network embedding, wireless network communication.

I. INTRODUCTION

WITH the continuous development and improvement of network technology and infrastructure, the communication mode between terminals gradually changes from wired communication to wireless network communication [1]–[7]. In the context of building a smart city, this transformation is

Manuscript received August 30, 2020; revised October 12, 2020; accepted October 25, 2020. Date of publication November 3, 2020; date of current version January 22, 2021. This work was supported in part by the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-006, and in part by “the Fundamental Research Funds for the Central Universities” of China University of Petroleum (East China) under Grants 20CX05017 A and 18CX02139 A. The review of this article was coordinated by Prof. Jelena Misić. (Corresponding authors: Neeraj Kumar and Peiyong Zhang.)

Peiyong Zhang and Chao Wang are with the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, P. R., China (e-mail: 25640521@qq.com; wangch_upc@qq.com).

Gagangeet Singh Aujla is with the Department of Computer Science, Durham University, Durham DH1 3LE, U.K. (e-mail: gagi_aujla82@yahoo.com).

Neeraj Kumar is with the School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand 248003, India, with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala 147111, India, and also with the Department of Computer Science and Information Engineering, Asia University Taiwan and King Abdul Aziz University, Jeddah 21577, Saudi Arabia (e-mail: neeraj.kumar@thapar.edu).

Mohsen Guizani is with the CSE Department, Qatar University, Doha 2713, Qatar (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/TVT.2020.3035341

particularly important. Wireless communication can overcome many inconveniences and defects of wired communication. For example, in the field of IoV, signal indication and resource allocation instructions are transmitted in the form of wireless signals. In the environment of IoV, the self driving vehicle needs to obtain road information and other environmental information independently [8]. This requires all kinds of network infrastructure (satellite, server, signal station, etc.) to provide network resources for vehicles in time to ensure their normal operation [9]–[11]. The transmission of these network resources needs a lot of bandwidth as a guarantee. Using wireless network communication to schedule network resources is the task of wireless communication [12], [13]. Considering the limitation of traditional network architecture in resource scheduling, it is necessary to design a reliable scheme of resource scheduling using wireless communication mode in virtual network environment [14]–[16].

In recent years, the emerging network virtualization (NV) can effectively make up for the shortcomings of existing networks and VNE is one of the key issues in NV research [17], [18]. Multi domain VNE is the focus of this field. According to the different needs of developers and the different goals of interests, the mapping goals will be different. There are mainly mappings with the minimum resource cost, the shortest time delay, the largest bandwidth, load balancing mapping and the largest revenue [19], [20].

The application of IoV shows the necessity and importance of using wireless network communication to realize network resource scheduling [21]. In the context of IoV, driver-less vehicles may need the support of different network resources (CPU, bandwidth, delay requirements, etc.) according to different space-time environments. Its application scenario is shown in the Fig. 1. For example, the car needs to know the road conditions in advance, judge the distance from other vehicles and perceive the change of indicator light. The acquisition of these information happens over the network and the network bandwidth resources play a great role in the transmission of these information. In a short time, there may be a large number of driver-less vehicles on the road, who need to obtain road information simultaneously. This situation requires that the wireless network must be robust enough to transmit enough resource scheduling information. This kind of instantaneous and large amount of network resource demand will bring huge pressure to

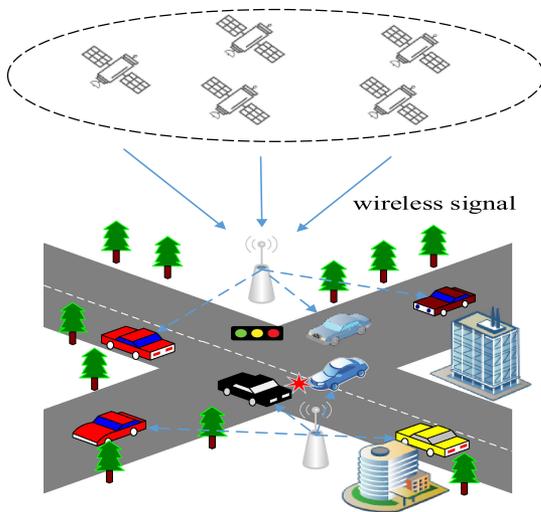


Fig. 1. Wireless communication in IoV scenario.

the underlying network infrastructure [22]–[24]. In this paper, we will focus on bandwidth resource allocation in wireless network environment.

For IoV and other services with high bandwidth requirements, it is of great significance to study the bandwidth index in multi domain VNE. How to realize the resource scheduling of VNR in wireless communication environment is an urgent problem. With the blowout growth of network end-users, the network scale has increased dramatically. So the substrate network also becomes more and more complex. The VNE problem on a single domain cannot meet the needs of modern business more and more. A good solution to this problem is to transform the single-domain VNE problem into the multi-domain VNE problem. BA-VNE algorithm focus on bandwidth constraints and use PSO to optimize. Good results are achieved in the final experiment.

A. Contributions

The main contributions of this article are listed below.

1) In order to meet the demand of network resources in wireless network communication mode, we design and implement a BA-VNE algorithm for IoV scenarios. We focus on the demand of IoV users for a large amount of bandwidth resources. According to the objective function, we select the link whose bandwidth value is greater than the average bandwidth in the domain to embed. In order to ensure adequate bandwidth resource supply.

2) In order to improve the performance of BA-VNE algorithm, we adopt a centralized hierarchical multi domain VNE architecture. The cooperation of local controller and global controller can improve the efficiency of cross domain problem solving. In addition, we add genetic mutation factor into PSO algorithm, which can effectively avoid the situation that particles fall into local optimum and cannot reach global optimum.

3) We have done a mass of simulation experiments to verify the effectiveness of BA-VNE algorithm. BA-VNE algorithm is compared with other representative algorithms in the aspects of link bandwidth selection, mapping cost, VNR acceptance rate

and link utilization. Experiments show that BA-VNE algorithm is effective.

B. Organization

The rest of the paper is organized as follows. Part II analyzes the research work of VNE problem. Part III introduces some related problems of multi-domain VNE. Part IV introduces the model of BA-VNE algorithm and puts forward the evaluation index. Part V introduces the BA-VNE algorithm and gives the pseudo-code implementation. Part VI introduces the setting of experimental environment and analyzes the experimental results. In the last part of the paper, we summarize the main content of the paper.

II. RELATED WORK

A. Centralized Multi-Domain VNE Algorithm

There are different classification criteria for multi-domain VNE algorithms [25]. We classify multi-domain VNE algorithms according to centralized architecture and distributed architecture. The BA-VNE algorithm is based on the centralized network architecture. The embedding process of the algorithm is completed by the global controller and the local controller [26]–[28].

Reference [29] proposed a multi-domain VNE scheme based on software-defined network (SDN). The mapping costs of nodes and links within the local controller are first estimated. Based on this cost, a candidate node selection algorithm is designed. Finally, the PSO algorithm is extended to the global controller to reduce the overall embedding cost. Therefore, the paper focuses on the influence of embedding cost on multi-domain VNE algorithm. Reference [30] proposed a cross-domain VNE algorithm based on minimum cost. The algorithm selects the virtual nodes to be mapped by pre-set constraints and selects the virtual links to be embedded by the minimum weight routing algorithm. Then the substrate path set with the least weight is selected by using Kruskal's minimum spanning tree and finally the embedding of virtual network is carried out. Therefore, the paper focuses on the influence of resource cost on the embedding algorithm of cross-domain virtual network. Reference [31] proposed a centralized cross-domain VNE algorithm. Based on dynamic topology perception and resource attributes, a network topology attribute evaluation model is established. The model is then used to measure the mapping priority of the nodes. Finally, the network load state is analyzed according to link resource cost. A meta heuristic algorithm based on iterative search is proposed in [32]. The algorithm mainly solves the complexity and scalability of multi-domain VNE. The main idea of the algorithm is to promote link embedding by node embedding. This can achieve cost effectiveness and help to decompose the network requests between cloud service providers in the network cloud environment online.

B. Distributed Multi-Domain VNE Algorithm

The distributed virtual network architecture is different from the centralized virtual network architecture. In the distributed

VNE architecture, each InP is an independent individual and the information of each physical domain is not public. This is due to factors such as confidentiality and commercial competition among InPs. In this case, the VNE work needs the cooperation of InPs and service providers (SP) [33]–[35].

In reference [36], a mapping strategy based on bidding method is proposed. The strategy first assigns the virtual network embedding requests (VNER) from each SP to the InP. The main task of the InP is to select the subgraph of the VNER to be embedded, then send the rest of the VNER to other InPs. Reference [37] proposed the V-Mart architecture based on the authenticity of virtual resource auction. This architecture uses two-step Vickrey model to build a virtual environment similar to the bidding platform for InPs. Through bidding and negotiation among participating InPs, VNR decomposition is carried out to solve the problem of multi-domain VNE. Reference [38] proposed a policy-based VNE architecture VINEA. The algorithm separates the policy (target) from the underlying mapping mechanism (resource, virtual network mapping) and can contain existing embedding methods. It can be used to design VNE solutions for different scenarios just by instantiating different policies. The purpose of this approach is to improve the utilization of the underlying network resources, thereby maximizing InPs revenue. In addition, reference [39] also proposed a VNE algorithm based on PSO. The algorithm uses SDN architecture to solve the problem of VNE which focuses on embedding cost. So to sum up, the existing multi-domain VNE algorithms, whether centralized or distributed, mainly focus on embedding cost, embedding cost, embedding success rate and other indicators. So far as we know, no multi-domain VNE algorithm has focused on bandwidth as a constraint.

III. PROBLEM SPECIFICATION

A. Description of the Basic Problem of Multi-Domain VNE

The problem of VNE can be divided into two parts: embedding in a single network domain and embedding across multiple underlying networks. The latter is more complex than the former. The complexity is reflected in the fact that the entire underlying infrastructure passes through multiple management domains or data centers. Multiple InPs work together to form the underlying infrastructure network, while different InPs manage their respective underlying nodes and links. Information between InPs is not disclosed to each other. Therefore, the pricing of link resources between multiple InPs is not uniform, which makes it more difficult to determine the bandwidth.

In the cross domain VNE architecture, the SP will send users' VNRs to the underlying network to request network resources. The global controller will segment the VNRs and distribute them to the local controller. Finally, the local controller will allocate the network resources needed for VNE to each network request. Given a VNR and the underlying infrastructure composed of multiple InP management domains, the embedding of the virtual network to the underlying infrastructure is accomplished with the goal of seeking the maximum bandwidth resource link. The InP does not publicly disclose link resource information within its managed domain, nor does it determine resource information

on inter-domain links. Therefore, the decision maker of the VNE does not have global information about the entire underlying infrastructure, which makes it difficult to determine the maximum bandwidth.

The underlying layer of the multi-domain VNE framework is composed of multiple substrate networks. A VNE request may be mapped to multiple substrate networks. Each underlying network is managed by a corresponding InP. Each underlying network has logic controllers (centralized or distributed) to perform management and control functions for infrastructure vendors. Each boundary node in the different underlying network is connected by inter-domain link to form the substrate network.

There is a limit to the amount of information that can be uploaded from the local controller to the global control. If all the node and link information of the underlying network is uploaded to the global controller, it will inevitably cause a lot of overhead. Therefore, only the following three kinds of information are uploaded to the global controller:

- 1) Available resources and unit price for candidate nodes (This part is described in part III-B).
- 2) Available bandwidth resources and unit price of links between physical domains.
- 3) Link bandwidth resource aggregation unit from candidate node to boundary node in the domain. That is, only boundary nodes are kept in a physical domain and the remaining nodes are replaced by a non-boundary node. All links are also aggregated between the boundary node and the non-boundary node. The in-domain link unit price is the link aggregation unit price.

Taking Fig. 2 as an example, link aggregation occurs in both physical domain 1 and physical domain 2 during information upload. For physical domain 1, since only boundary nodes and one non boundary nodes are reserved, only nodes *B*, *C* and *D* are uploaded to the global controller. Before uploading the information, there is a link connection between node *A*, node *B* and node *C*. The upper bandwidth are 4 and 6. Because of the link aggregation, the two links originally connected to node *A* are aggregated into one link. The bandwidth value of the link also changes from the sum of the two aggregated link bandwidth values to 10. Physical domain 2 also has link aggregation when uploading information. Only boundary nodes *F* and *H* and non boundary nodes *G* are reserved.

B. Selection of Candidate Nodes

The specific steps for selecting candidate nodes are as follows:

- 1) Select the boundary nodes in the candidate domain as the candidate nodes until all boundary nodes are selected.
- 2) Select the non boundary nodes with substrate link with the boundary nodes as the candidate nodes. The substrate link between the two nodes is saved and the bandwidth value on the link is recorded at the same time.
- 3) If there is no link between the non boundary node and the boundary node, continue to select the next hop node of the boundary node.

Repeat the above steps until all nodes in the candidate domain are selected. Finally, the bandwidth values of all links in the

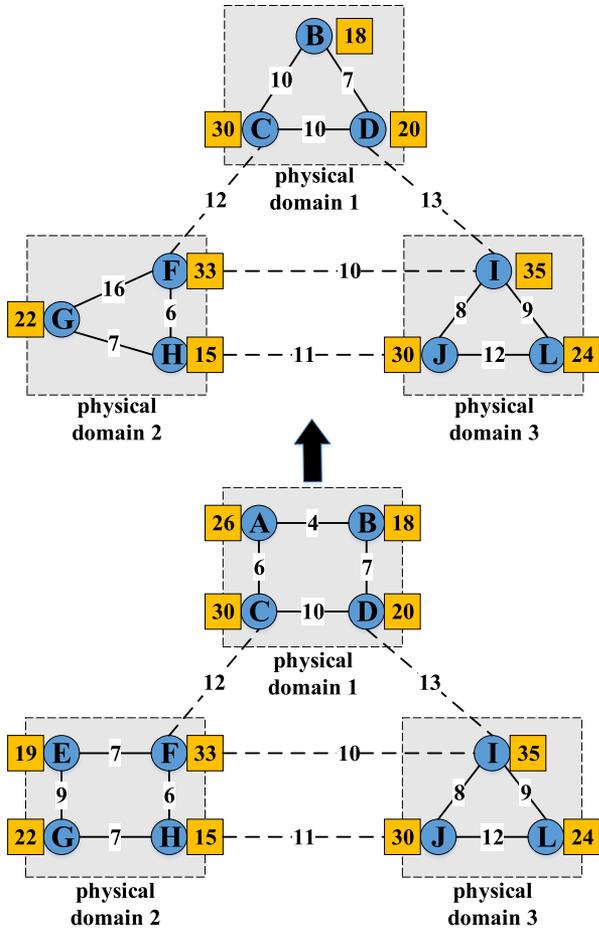


Fig. 2. Local controller uploads information.

domain are recorded.

$$\text{bandwidth}(n_{border}^s, n_i^s) \neq 0. \quad (1)$$

The selection condition is shown in Eq. (1). As long as the link bandwidth value between the boundary node and a node is not 0, there is a substrate link between the two nodes. Then both nodes can be used as candidate nodes and the substrate link between them can be saved.

IV. NETWORK MODELS AND EVALUATION INDICATORS

A. Underlying Network Model

The underlying network model can be represented by $G^s = \{N^s, L^s\}$, specifically, this is a weighted undirected graph. G^s represents the entire underlying network. N^s represents all the underlying network nodes and L^s represents all the underlying network links. The computing power of the substrate node $n^s \in N^s$ is expressed in terms of $CPU(n^s)$. The substrate link between the substrate nodes x and y is represented by $l^s(x, y)$, where $l^s(x, y) \in L^s$. The bandwidth capacity of the substrate link is expressed in terms of $BW(l^s)$.

Fig. 3(c) is an example diagram of a substrate network. The circles represent the substrate nodes. The amount of computational resources available for the substrate node is represented by

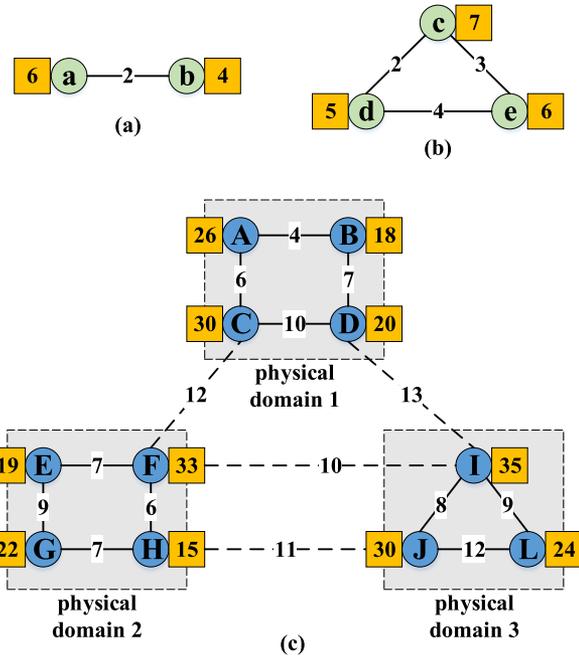


Fig. 3. Diagram of virtual network and substrate network.

the number in the rectangle. The connections between the nodes represent substrate links. The amount of bandwidth available for the substrate link is represented by the number of connections between the substrate nodes.

B. Virtual Network Model

The virtual network model can be represented by $G^v = \{N^v, L^v\}$ which is also a weighted undirected graph. Where G^v represents the entire virtual network, N^v represents all the virtual network nodes and L^v represents all the virtual network links. The computing resource requirements of virtual node $n^v \in N^v$ is expressed in terms of $CPU(n^v)$. The virtual link between the virtual nodes x and y is represented by $l^v(x, y)$, where $l^v(x, y) \in L^v$ has a bandwidth requirement of $BW(l^v)$.

Both (a) and (b) in Fig. 3 are examples of virtual networks. The circle represents the virtual node. The computing resource requirements of virtual nodes are represented by numbers in the rectangle. The lines between the nodes represent virtual links. The bandwidth resource demand of virtual link is represented by the number of the connection between virtual nodes.

C. VNR Model

The VNR first arrives at the global controller and subnet partition is performed under the operation of the global controller. A virtual node is divided into a separate subgraph of cells. The VNR can be represented as $G^v = \{N^v, L^v, C_N^v, C_L^v, D_N^v\}$. Where G^v represents the entire VNR process. N^v represents all the virtual network nodes and L^v represents all the virtual network links. C_N^v represents the computational footprint of the virtual node, which is the CPU footprint. C_L^v represents the bandwidth resource demand of the virtual link. D_N^v represents the candidate domain of the virtual node.

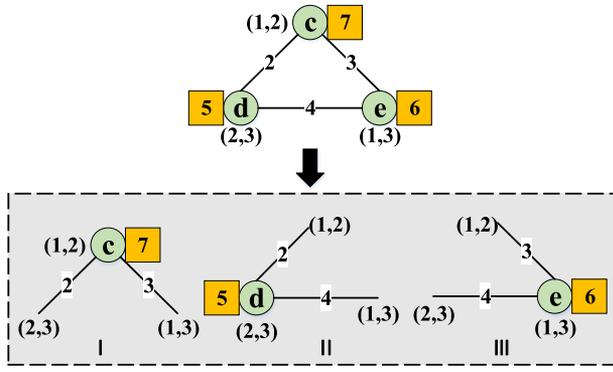


Fig. 4. Virtual network requests and their subgraphs.

VNRs and their subgraphs are shown in Fig. 4. The virtual network with three virtual nodes and three virtual links is divided into three VNR subgraphs. The subgraph includes information such as the computation resource demand, the candidate domain number of virtual node and the bandwidth demand of virtual link.

Take the example of the VNR subgraph I. The candidate domains of virtual node c include physical domain 1 and physical domain 2 which means that virtual node c can only select these two physical domains as candidate domains. The two virtual links connected to the node correspond to the candidate domains of virtual node d and virtual node e respectively. The other two virtual nodes are similar.

D. The Objective Function

In order to ensure that all virtual nodes can be embedded, we need to select the substrate links with the largest remaining bandwidth resources to map to meet the service requirements of the IoV. Here, we define a mapping target formula:

$$BW[l^s(u, v)] \geq \frac{\sum L^s BW(l_i^s)}{\text{count} + 1}. \quad (2)$$

We initialize the substrate network before selecting the candidate nodes. Select the substrate links that meet the standards and the selection criteria are shown in Eq. (2). First, sum up the remaining bandwidth resources of all links in a physical domain to obtain the total bandwidth resources of the links in the domain. Then the total link bandwidth resource is divided by the number of links in the domain and the mean value is taken as a selection criterion. It can be considered that if the bandwidth value of a link is greater than the average value, then the link is eligible. Where, the denominator *count* represents the number of substrate links, plus 1 is to prevent the situation that the denominator is 0. The numerator is the sum of the bandwidth of all links in the domain. On the left side of the formula is the bandwidth value on the link between the substrate nodes u and v .

In Fig. 4 of the VNR subgraph I and Fig. 2 by the local controller in the upload of the substrate network as an example, shows the specific meaning of the Eq. (2). Suppose virtual node c selects substrate node A in candidate domain 1 for mapping. Since substrate node A is not uploaded, it should be calculated through substrate node B at this time. First, the bandwidth mean

value $\frac{10+10+7}{3}$ in candidate domain 1 is calculated, then the mean value is compared with the value of all link bandwidth in that domain. The bandwidth value of the link between substrate node B and C is 10. The bandwidth value of the link between substrate node C and D is also 10. The bandwidth value of the two links is larger than the average value, so they can be used as candidate links. However, the bandwidth value of the link between substrate node B and D is 7, which is less than the average value, so this link cannot be used as a candidate link. In this form, the local controller passes all substrate nodes and links to boundary nodes that meet the bandwidth value requirements up to the global controller. This completes the selection of candidate nodes. Eq. (2) shall meet the following conditions:

$$\max[CPU(n^s) \geq CPU(n^v)], \quad (3)$$

$$\max[BW(l^s) \geq BW(l^v)], \quad (4)$$

$$\text{num}(n_v^s) = 1, \quad (5)$$

$$D(n^s) = D(n^v), \quad (6)$$

$$n_{v_1}^s = n_{v_2}^s \text{ if } v_1 = v_2, \quad (7)$$

$$p(n_i^s, n_k^s) = p(n_k^s, n_i^s). \quad (8)$$

Among them, Eq. (3) indicates that the computing resource of substrate node n^s should not be less than that of virtual node n^v . Eq. (4) indicates that the bandwidth resource of substrate link l^s should not be less than that of virtual link l^v . Eq. (5) indicates that a virtual node n^v in a VNR can only be embedded in one substrate node n^s . Eq. (6) indicates that the target substrate node of the virtual node must be within the candidate domain. Eq. (7) indicates that a substrate node can only be mapped by one virtual node. Eq. (8) indicates that the path of the substrate link has no directionality.

E. The Evaluation Index

In this part, we formulate the evaluation index of the VNE algorithm.

InPs and SPs need to be charged according to the underlying network resources consumed by users. In the problem of VNE, VNRs (user function requirements) are mapped to the underlying network, which mainly consumes CPU resources and bandwidth resources. Therefore, InPs and SPs charge users mainly according to the amount of CPU resources and bandwidth resources consumed. According to the resource unit price set by InPs and SPs, the VNE cost is expressed as follows:

$$\text{Cost} = \sum_{\text{num}(n^v)} CPU(n^v) \times \text{Cost}(n_v^s) + \sum_{\text{num}(l^v)} BW(L^v) \times \text{Cost}(l_v^s), \quad (9)$$

where, $CPU(n^v)$ represents the computing resource demand of virtual node n^v . $BW(L^v)$ represents the bandwidth resource demand of virtual link l^v . n_v^s represents that the substrate node of the final mapping of virtual node n^v is n^s . l_v^s represents that virtual link l^v is mapped to substrate link set l^s . Because a virtual link can be split and mapped to different substrate

links. $Cost(n_v^s)$ represents the computing resource unit price of substrate node n^s . $Cost(l_v^s)$ represents the bandwidth resource unit price of substrate path l^s .

The VNE request acceptance rate is shown as follows:

$$Acp = \lim_{N \rightarrow \infty} \frac{\sum_{n=1}^N Acp(G^V, n)}{\sum_{n=1}^N All(G^V, n)}. \quad (10)$$

The acceptance rate of VNRs refers to the proportion of successful mapping of VNRs. Where n represents the number of virtual networks. The ratio of the number of successful virtual networks $\sum_{n=1}^N Acp(G^V, n)$ to the total number of VNRs that arrive $\sum_{n=1}^N All(G^V, n)$.

Link utilization is shown as follows:

$$LinkUse = \lim_{N \rightarrow \infty} \frac{\sum_{n=1}^N Num(L^V \rightarrow L^S)}{All(L^S)}. \quad (11)$$

Link utilization refers to the proportion of mapped substrate links to all substrate links. The ratio of the number of successful virtual links $\sum_{n=1}^N Num(L^V \rightarrow L^S)$ to the total number of links $All(L^S)$ in the substrate network.

V. ALGORITHM DESCRIPTION AND IMPLEMENTATION

A. Algorithm Description

We divide the implementation of BA-VNE algorithm into the following steps.

1) The specific steps of candidate node selection are described in section III-B. We take the pre mapping cost of virtual nodes as the selection criteria of candidate nodes. The pre mapping cost of node can be expressed as follows:

$$Cost_{forecast} = CPU(n^v) \times Cost(n_v^s). \quad (12)$$

2) VNR pre-mapping: The local controller sends the selected set of candidate nodes to the global controller. The global controller pre maps the VNR based on this information. If the premapping is successful, the premapping results are delivered to the local controller.

3) Intra-domain embedding of VNRs: The local controller receives the pre-mapping result from the global controller, then carries out intra-domain node mapping. This process includes the embedding of virtual nodes and links. After the virtual node is successfully embedded, the intra-domain link with the most remaining bandwidth resources around the node is selected for embedding.

4) Inter-domain link embedding: If all intra-domain embeds are successful, the global controller completes the inter-domain link embedding. The basis is to first select the substrate link with the most remaining bandwidth resources for embedding. Finally, the embedding process is finished.

The flowchart of BA-VNE algorithm is shown in Fig. 5.

B. Algorithm Implementation

In this part, we will describe the implementation process of two important parts of BA-VNE algorithm in detail. These two steps are candidate nodes selection and VNR pre mapping. We give their pseudo code implementation.

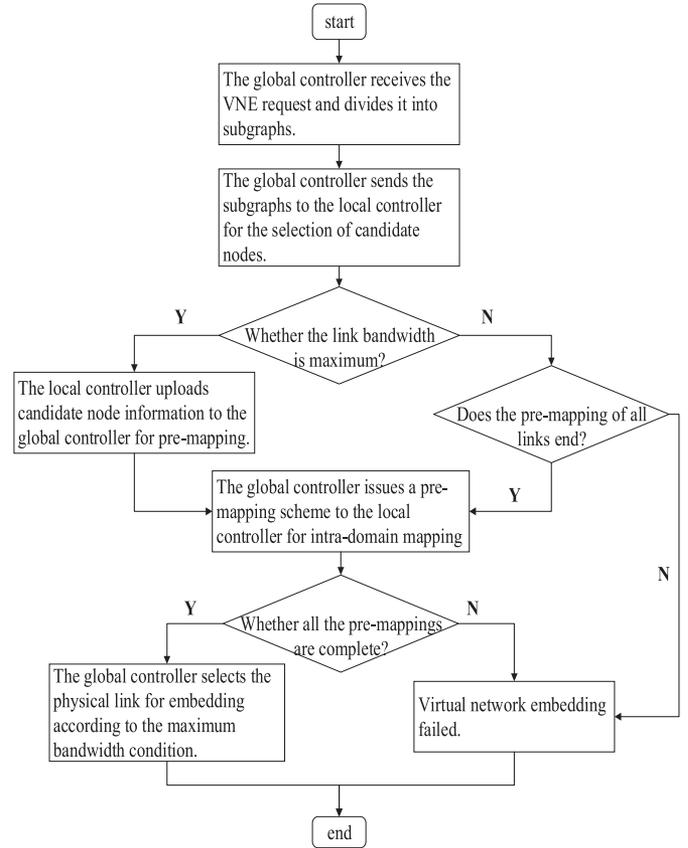


Fig. 5. BA-VNE algorithm flowchart.

1. *Candidate node selection algorithm*: The second line of the algorithm represents setting the substrate nodes to a state where none of them are mapped, which is initialization. Because one substrate node can only host one virtual node from the same VNR, if the substrate node has been mapped by a virtual node in a VNR, initialization must be carried out to prevent affecting the node mapping in other VNRs. Line 4 represents the node loop operation (if there are boundary nodes, judge them first). Line 5 first determines whether a link exists between two nodes. If so, line 6 determines whether the remaining bandwidth resource found between two nodes is greater than the average link bandwidth of the physical domain. Because the problem to be studied is multi-domain VNE, inter-domain link embedding is necessary, so the boundary nodes must be embedded. Line 7 indicates that the selected candidate node is saved.

2. *VNR pre-mapping*: After the global controller receives the candidate node information (including the case where it is connected to the boundary node), this information is combined with inter-domain links to form a new network topology called global candidate node network. Based on this network, PSO is used to realize the premapping of multi-domain VNE scheme with maximum bandwidth. PSO is a typical natural heuristic algorithm, which is based on the analogy of the foraging process of birds. It is often used to solve global search problems to get the optimal solution [40]–[42]. In the discrete PSO algorithm, the calculation formula of particle position and velocity is as

Algorithm 1: Candidate Node Selection Algorithm.

```

1: Input:  $G_i^s = \{N_i^s, L_i^s\}, G_i^v = \{N_i^v, L_i^v\}$ ;
2: Output:  $N_{i,candidate}^s$ ;
3: procedure
4:    $\phi(N^s) \leftarrow 0$ ;
5: end procedure
6: for all  $n_i^s \in N_i^s$ 
7:   if  $bandwidth(n_{border}^s, n_i^s) \neq 0$ 
8:     if  $BW[l^s(u, v)] \geq ave\_bandwidth$ 
9:        $n_{candidate}^s = (n_{border}^s, n_i^s)$ ;
10:    end if
11:  end if
12: end for
13: return  $n_{candidate}^s$ ;

```

follows:

$$v_i^{new} = v_i + c_1 r_1 (x_i^{pb} - x_i) + c_2 r_2 (x^{gb} - x_i), \quad (13)$$

$$x_i^{new} = x_i + v_i^{new}. \quad (14)$$

There is a concept of fitness function in PSO, which we express as Eq. (2). v_i originally represents the speed of particles in PSO. In BA-VNE algorithm, v_i represents the change direction of the i th mapping scheme. c_1 and c_2 represent the learning factors in PSO algorithm. Reasonable setting of the learning factors can improve the efficiency of the algorithm. r_1 and r_2 represent random numbers between 0 and 1, which are variable factors. x_i^{pb} is meant to represent the best position of a particle in PSO. In BA-VNE algorithm, the fitness function representing this mapping scheme is optimal. x^{gb} is meant to represent the global optimal position in PSO. In BA-VNE algorithm, the fitness function of this mapping scheme is the best among all particle mapping schemes.

The steps of PSO are as follows:

- 1) *Initialize*: Location and velocity of randomly generated particles to produce an embedding scheme.
- 2) *Evaluate each particle*: Calculate the fitness value of the particle and the embedding cost of the embedding scheme.
- 3) *Update*: Velocity & position of particle \rightarrow Eqs. (13–14).
- 4) *Check whether the end conditions are met*: If the maximum number of iterations has been reached, PSO algorithm will stop and output the optimal solution. Otherwise, go back to step (2) and continue.

The second line of the algorithm represents setting the candidate substrate node to a state that is not embedded, that is, initialization. Line 4 represents looping through the maximum number of iterations. Line 5 is a loop for the number of particles. Line 6 updates the position of the particle. Line 7 is the update particle velocity. Lines 8 to 12 are the judgment and update operation of particle position.

C. Algorithm Complexity

The complexity of BA-VNE algorithm is mainly caused by several main processes, such as candidate node selection, PSO and VNE process. Assume that the number of virtual nodes

Algorithm 2: Virtual Network Request Premapping.

```

1: Input:  $G^v = \{N^v, L^v\}, PseudoTopo$ ;
2: Output:  $premapping(n^v)$ ;
3: procedure
4:    $\phi(N^s) \leftarrow 0$ ;
5: end procedure
6: for  $i < Max\ Iteration$ 
7:   for  $j < Partial\ Number$ 
8:      $X_{i+1} = Update(X_i)$ ;
9:      $V_{i+1} = Update(V_i)$ ;
10:    if  $Cost(X_{i+1}) < Cost(X^{pb})$ 
11:       $X^{pb} = X_{i+1}$ ;
12:    end if
13:    if  $Cost(X^{pb}) < Cost(X^{gb})$ 
14:       $X^{gb} = X^{pb}$ ;
15:    end if
16:  end for
17: end for
18: return  $premapping(n^v)$ ;

```

in the VNR is N . The number of physical nodes in the whole underlying network is M . The number of physical domains is D . In PSO algorithm, the number of particles is P and the number of iterations is I . Then the complexity of BA-VNE algorithm is calculated as follows:

$$\begin{aligned}
O(N, M, D, P, I) &= O(N \cdot M \cdot N \cdot D) \\
&\quad + O(P \cdot I \cdot N \cdot D \cdot N \cdot D) \\
&= O(N^2 \cdot M \cdot D) + O(N^2 \cdot D^2 \cdot P \cdot I). \quad (15)
\end{aligned}$$

Because the number of underlying network domains and the number of physical nodes are fixed during the VNR process, the complexity of the BA-VNE algorithm can be expressed as $O(N^2)$.

VI. PERFORMANCE EVALUATION

This chapter first introduces the environment of simulation experiment and the parameter setting of simulation experiment. Then the experimental results are presented and analyzed. We compare the BA-VNE algorithm proposed in this paper with VNE-PSO algorithm proposed in reference [29], MC-VNM algorithm proposed in reference [30], LID-VNE algorithm proposed in reference [43] and MP-VNE algorithm proposed in reference [44] to verify the performance of the algorithm. The main ideas are listed in Table I. Specifically, we compare bandwidth selection, mapping cost and VNR acceptance rate.

A. Experiment Environment and Parameter Setting

The simulation experiment was done on a computer with 8 GB memory, 64-bit width and win8 operating system. We use traditional GT-ITM tools to generate the required network topology. The analysis of the experimental results and the drawing

TABLE I
ALGORITHM IDEAS

Notation	The algorithm description
BA-VNE	The candidate nodes are selected according to the maximum bandwidth. PSO with genetic variation factors was used to solve the node embedding scheme.
VNE-PSO [29]	The hop number from the boundary node is used as the selection criterion of candidate nodes. PSO algorithm is used to optimize the VNE scheme and the best VNE scheme is used as the final algorithm.
MC-VNM [30]	Based on the minimum mapping cost, the virtual link is mapped first, then the virtual node is mapped.
LID-VNE [43]	The bandwidth requirement is taken as the VNE index. Each network domain performs node mapping and link mapping according to different bandwidth requirements.
MP-VNE [44]	A multi-objective VNE scheme transforms the multi-objective problem into a single objective problem. PSO algorithm is used to optimize the mapping scheme.

TABLE II
SUMMARY OF PARAMETER SETTINGS

Attribute name	Attribute value
number of physical domains	4
number of physical nodes	30
CPU resources of substrate nodes	U[100, 300]
CPU resource unit price	U[1, 10]
bandwidth resources	U[1000, 3000]
bandwidth resource unit price	U[1,10]
delay of substrate link	U[1,10]
number of boundary nodes in physical domain	2
connection rate	50%
price of inter-domain link bandwidth resource	U[5, 15]
interdomain link delay	U[10, 30]
number of candidate fields for each virtual node	2
CPU resource requirements	U[1,10]
bandwidth resource requirements	U[1,10]
connection rate	50%

of the line chart were completed by Origin 8.5. The important parameters in the network topology are set as follows:

In order to compare the experimental results in different numbers of data fields, we set up a total of four physical fields. The number of nodes in each physical domain is 30 and there are 2 boundary nodes in each physical domain. The amount of resources available for each substrate node is uniformly distributed [100, 300]. The unit price of substrate node resources obeys the uniform distribution of [1, 10]. The initial value of available bandwidth resources of the substrate link is set between 1000 and 3000. Link bandwidth resource unit prices follow uniform distribution [1, 10]. The substrate link delay obeys the uniform distribution of [1, 10]. The unit prices of inter-domain link bandwidth resources follow uniform distribution [5, 15]. Inter-domain link delay obeys uniform distribution [10, 30].

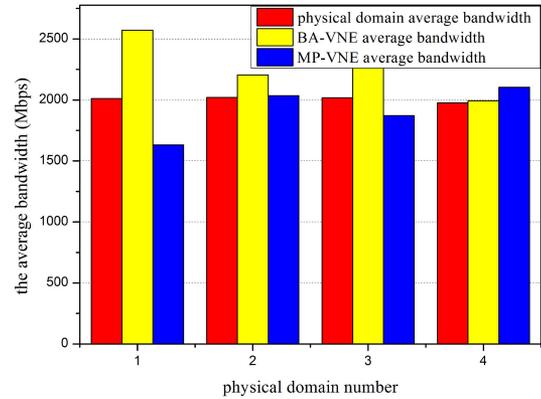


Fig. 6. Average bandwidth value.

Substrate links are connected between substrate nodes with a 50% probability.

The candidate fields for each virtual node are set to 2. CPU demand for virtual nodes is evenly distributed [1, 10]. The bandwidth resource demand of the virtual link is uniformly distributed [1, 10]. The node connection probability in the VNR is set to 50%. The arrival process of VNRs simulates Poisson distribution. The summary of simulation parameters is shown in Table II.

B. Experimental Results and Analysis

BA-VNE algorithm is essentially a VNE algorithm, so it is necessary to use the typical indicators (mapping cost, VNR acceptance rate, etc.) to evaluate the performance of the algorithm. Because BA-VNE algorithm focuses on the bandwidth factor, it also needs to compare with other algorithms in bandwidth. We designed five comparative experiments to evaluate the performance of BA-VNE algorithm. The first experiment is to prove the effectiveness of BA-VNE algorithm in selecting the largest possible bandwidth link. The following four experiments prove that the BA-VNE algorithm can guarantee the maximum bandwidth in terms of the synthetic mapping cost, VNR acceptance rate, average mapping delay and link utilization. The performance in these indicators is also excellent.

Experiment 1: Compare the average bandwidth with MP-VNE algorithm.

The purpose of this experiment is to show that BA-VNE algorithm has obvious advantage over other algorithms in choosing substrate links with large bandwidth for embedding. The average bandwidth of the two algorithms was compared in four physical domains. We also show the average bandwidth in each physical domain. (The average bandwidth of all links is obtained from the right side of Eq. (2).) The comparison result is shown in Fig. 6. The average link bandwidth selected by BA-VNE algorithm is significantly higher than the average link bandwidth selected by MP-VNE algorithm, while the MP-VNE algorithm has randomness in the selection of embedded link bandwidth.

Analysis of Experimental Results: The BA-VNE algorithm focuses on the index of maximum link bandwidth. With Eq. (2) as the selection criterion, only substrate links whose bandwidth

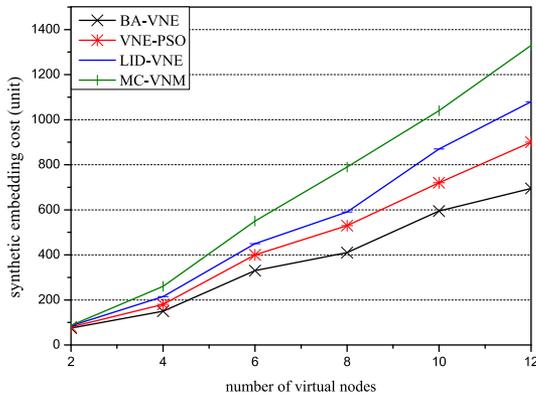


Fig. 7. Synthetic embedding cost.

value is greater than the mean value of substrate links can be embedded. Therefore, the value of substrate link bandwidth selected by BA-VNE algorithm is generally the largest. The MP-VNE algorithm uses weights to transform the multi-objective optimization problem into a single-objective optimization problem. Select the boundary node using the estimated mapping cost formula. So it does not focus on bandwidth maximization. Therefore, the experimental results of MP-VNE algorithm are random when the average bandwidth is taken as the evaluation standard. It is not guaranteed that the link with the maximum bandwidth will be selected for embedding in every VNE process.

Experiment 2: Comparison of algorithm synthetic embedding cost.

In the test synthetic embedding cost experiment, we keep changing the number of virtual nodes. Change the 4 virtual nodes we set up earlier to 2, 4, 6, 8, 10, and 12 virtual nodes. As the number of virtual nodes increases, the total cost of various embedding algorithms will inevitably increase. The BA-VNE algorithm is optimal. It can be seen from the Fig. 7 that the algorithm has the lowest comprehensive embedding cost in the case of various virtual nodes. Then comes the VNE-PSO algorithm, LID-VNE algorithm and the MC-VNM algorithm with the highest comprehensive embedding cost.

Analysis of Experimental Results: BA-VNE algorithm not only selects the maximum bandwidth link for mapping, but also takes into account the impact of mapping cost on the performance of the algorithm. Before mapping virtual nodes, the cost of mapping is estimated. By sorting the candidate nodes, the substrate nodes with the lowest estimated mapping cost are selected for embedding. This can reduce the mapping cost to some extent. VNE-PSO algorithm is worse than that of BA-VNE algorithm. Although the algorithm considers the cost estimation of node and link embedding, it selects the boundary node from the node closest to the boundary node. So this algorithm doesn't make a very good decision. LID-VNE is based on the distributed VNE architecture. Due to the business competition among SPs, they don't intend to disclose the details of network topology in their respective domains. Therefore, VNRs can only be embedded by randomly selecting substrate nodes and links, which leads to high mapping cost. The reason why the total mapping cost of MC-VNM algorithm is the highest is that the algorithm

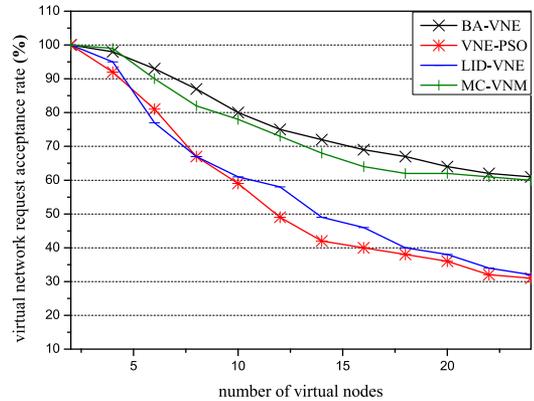


Fig. 8. Virtual network request acceptance rate.

doesn't use the traditional natural heuristic algorithm to solve the problem of VNE. The natural heuristic algorithm has been proved to be an effective method to solve NP-hard problems. MC-VNM algorithm simply uses the greedy strategy to select nodes and links to map, so the effect is the worst.

Experiment 3: VNR acceptance rate comparison test.

With the increase of the number of virtual nodes, the acceptance rate of VNE requests of the four algorithms presents a downward trend. However, the virtual request acceptance rate of BA-VNE and MC-VNM is obviously better than that of VNE-PSO and LID-VNE. The experimental results are shown in Fig. 8. In the MC-VNM and BA-VNE algorithms, the receiving rate of VNRs will stabilize around 60%, while VNE-PSO and LID-VNE algorithms will drop to around 30%.

Analysis of Experimental Results: With the continuous consumption of the underlying network resources, the acceptance rate of VNRs of all the algorithms is constantly decreasing. The VNR acceptance rate of the BA-VNE algorithm is higher than that of the other three algorithms, which benefits from the idea of load balancing while optimizing the bandwidth index. In order to as far as possible in the case of no loss of bandwidth index, but also as good as possible reception rate. The MC-VNM algorithm uses the Kruskal spanning tree idea to select the link-first embedding with the minimum link unit price. The node embedding scheme is determined by the link embedding scheme, which also has a high VNR acceptance rate. The reason why VNE-PSO algorithm and LID-VNE algorithm have low acceptance rate of VNRs is that the methods of selecting substrate nodes are too rigid. In the process of VNE, there are often insufficient node resources or link resources. Therefore, the performance of these two algorithms is poor.

Experiment 4: Average embedding delay comparison experiment.

The experimental results of mean embedding delay are shown in Fig. 9. Taking the number of nodes requested by the virtual network as the independent variable, the average embedding delay of the four algorithms increases with the number of virtual nodes. The average delay of MC-VNM algorithm is the largest, reaching above 850. Next up are LID-VNE algorithm and VNE-PSO algorithm, both with a delay of 600 or more. Our BA-VNE algorithm has the lowest latency, below 500.

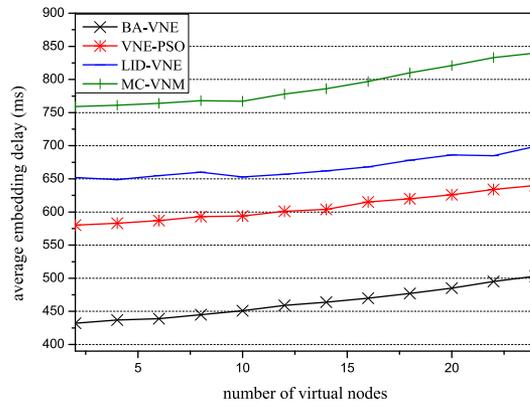


Fig. 9. Average embedding delay.

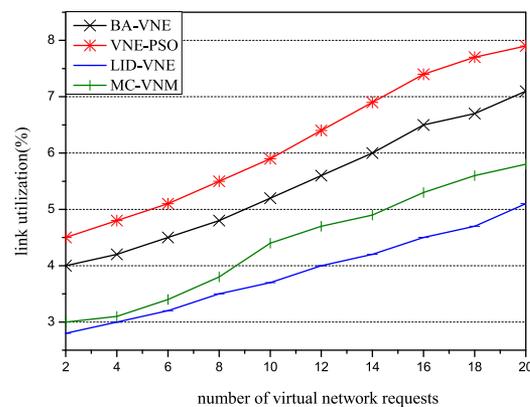


Fig. 10. Link utilization.

Analysis of Experimental Results: The difference of delay is mainly reflected in the choice of link mapping. First of all, LID-VNE algorithm adopts distributed VNE architecture. The disadvantages of using this architecture have been mentioned in the analysis of Experiment 3. VNE-PSO algorithm only selects boundary nodes as candidate nodes, which limits the possibility of selecting other substrate links. The virtual link can only select the substrate link connected with the boundary node for mapping and a single link selection may produce a large mapping delay. BA-VNE algorithm selects the link with the largest bandwidth for mapping and uses PSO algorithm for optimization. Experimental results show that this method has achieved good results. As shown in Fig. 9, the experimental effect of MC-VNM algorithm is the worst. Different from the other three algorithms, this algorithm does not use efficient heuristic method as a solution. It uses a relatively inefficient greedy strategy to complete the VNE problem, so the experimental results are poor.

Experiment 5: Link utilization comparison experiment.

Link utilization is an important index to evaluate the performance of VNE algorithm. The experiment we designed takes the number of VNRs as an independent variable and the experimental results are shown in Fig. 10. With the increase of the number of VNE requests, the link utilization of various algorithms increases. VNE-PSO algorithm has the highest link utilization rate, which can reach about 8%. The second is BA-VNE algorithm

and MC-VNM algorithm. The lowest link utilization is the LID-VNE algorithm, which is only 5%.

Analysis of Experimental Results: The difference of link utilization is caused by the link embedding strategy. VNE-PSO, BA-VNE and MC-VNM all adopt the shortest path method to embed virtual links, while LID-VNE adopts the multi-commodity flow method to embed virtual links. Experimental results show that the algorithm using shortest path strategy for link embedding has higher average link utilization. Because BA-VNE algorithm only selects links with larger link bandwidth for embedding, the range of embedded links is reduced, but VNE-PSO does not pay attention to this point. Therefore, the link utilization of BA-VNE algorithm is lower than that of VNE-PSO algorithm.

VII. CONCLUSION

Wireless network communication mode will be the leading role of network services in the future. Based on the application background of IoV in wireless network, this paper introduces a new network resource allocation method. With the development of NV technology, virtual network architecture will be the infrastructure of future network. In order to meet the demand for bandwidth resources in IoV, this paper proposes a BA-VNE algorithm. The algorithm focuses on the bandwidth constraints in network resources and allocates as large bandwidth links as possible for VNRs by setting bandwidth resource selection conditions. We use PSO algorithm to optimize the VNE algorithm. Especially, the mutation factor is added to the traditional PSO algorithm, which can effectively avoid the final result falling into the local optimum. Finally, we compare BA-VNE algorithm with other representative algorithms in many aspects. The simulation results show that the multi-domain VNE scheme proposed in this paper cannot only select the substrate link with the most remaining bandwidth resources for embedding. Moreover, it also performs well in terms of embedding cost, VNR acceptance rate and other aspects while ensuring the maximum bandwidth. Therefore, the VNE scheme with the largest bandwidth can solve the practical problems such as the IoV, video data backup and so on. It has good application significance.

REFERENCES

- [1] Q. Zhang, R. Wang, J. Yang, A. Lewis, F. Chiclana, and S. Yang, "Biology migration algorithm: A new nature-inspired heuristic methodology for global optimization," *Soft Comput.*, vol. 23, no. 16, pp. 7333–7358, Aug. 2018.
- [2] P. Zhang, H. Yao, M. Li, and Y. Liu, "Virtual network embedding based on modified genetic algorithm," *Peer-to-Peer Netw. Appl.*, vol. 2, pp. 1–12, Aug. 2017.
- [3] X. Zhang, "Research on key technologies of big data security and privacy protection," *Modern Trade Ind.*, vol. 40, no. 32, pp. 146–147, 2019.
- [4] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. P. C. Rodrigues, "SDN-enabled multi-attribute-based secure communication for smart grid in IIoT environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2629–2640, Jun. 2019.
- [5] S. Saharan, S. Bawa, and N. Kumar, "Dynamic pricing techniques for intelligent transportation system in smart cities: A systematic review," *Comput. Commun.*, vol. 150, pp. 603–625, 2020.
- [6] P. Zhang, H. Li, Y. Ni, F. Gong, M. Li, and F. Wang, "Security aware virtual network embedding algorithm using information entropy TOPSIS," *J. Netw. Syst. Manag.*, vol. 5, pp. 1–23, 2019.

- [7] R. K. Chahal, N. Kumar, and S. Batra, "Trust management in social Internet of Things: A taxonomy, open issues, and challenges," *Comput. Commun.*, vol. 150, pp. 13–46, 2020.
- [8] A. Kumari, R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "When blockchain meets smart grid: Secure energy trading in demand response management," *IEEE Netw.*, vol. 34, no. 5, pp. 299–305, Sep./Oct. 2020.
- [9] H. Li, W. Li, H. Wang, and J. Wang, "An optimization of virtual machine selection and placement by using memory content similarity for server consolidation in cloud," *Future Gener. Comput. Syst.*, vol. 84, pp. 98–107, 2018.
- [10] P. Zhang, C. Wang, Z. Qin, and H. Cao, "A multidomain virtual network embedding algorithm based on multiobjective optimization for Internet of Drones architecture in Industry 4.0," *Softw. Pract. Experience*, pp. 1–19, 2020, doi: <https://doi.org/10.1002/spe.2815>.
- [11] W. Li, Z. Chen, X. Gao, W. Liu, and J. Wang, "Multimodel framework for indoor localization under mobile edge computing environment," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4844–4853, Jun. 2018.
- [12] J. Wang, Y. Gao, X. Yin, F. Li, and H. J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, 2018.
- [13] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H. J. Kim, "Energy efficient routing algorithm with mobile sink support for wireless sensor networks," *Sensors*, vol. 19, no. 7, pp. 1–19, 2019.
- [14] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on computing, network, and storage resource constraints," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3298–3304, Oct. 2018.
- [15] R. Matthias and S. Schmid, "Virtual network embedding approximations: Leveraging randomized rounding," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 2017–2084, Oct. 2019.
- [16] 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. 1, pp. 1–9, 2016.
- [17] D. Zhao, "Highly virtualized IP networks," *Commun. World*, vol. 26, no. 8, pp. 66–67, 2019.
- [18] R. Chaudhary and N. Kumar, "LOADS: Load optimization and anomaly detection scheme for software-defined networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12329–12344, Dec. 2019.
- [19] Y. Zhu, "Research on embedding key technologies across multi-domain virtual networks," *Univ. Electron. Sci. Technol. China*, vol. 3, pp. 5–8, 2014.
- [20] T. Mano, T. Inoue, D. Ikarashi, K. Hamada, K. Mizutani, and O. Akashi, "Efficient virtual network optimization across multiple domains without revealing private information," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 477–488, Sep. 2016.
- [21] X. Li, Y. Wang, P. Vijayakumar, D. He, N. Kumar, and J. Ma, "Blockchain-based mutual-healing group key distribution scheme in unmanned aerial vehicles Ad-Hoc network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11309–11322, Nov. 2019.
- [22] J. Zhao, Z. Hu, B. Xiong, and K. Li, "Accelerating packet classification with counting bloom filters for virtual openflow switching," *China Commun.*, vol. 15, no. 10, pp. 117–128, 2018.
- [23] J. Zhang, D. Zhang, K. Xie, H. Qiao, and S. He, "A VMIMO-based cooperative routing algorithm for maximizing network lifetime," *China Commun.*, vol. 14, no. 4, pp. 20–34, 2017.
- [24] N. Kumar, G. S. Aujla, A. K. Das, and M. Conti, "ECCAuth: A secure authentication protocol for demand response management in a smart grid system," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6572–6582, Dec. 2019.
- [25] 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, 2018.
- [26] S. Zhang, Z. Qian, and J. Wu, "Virtual network embedding with opportunistic resource sharing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 816–827, Mar. 2013.
- [27] 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, Jan. 2017.
- [28] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1621–1631, Jun. 2019.
- [29] R. Geng and H. Lu, "Multi-domain SDN virtual network mapping algorithm," *A Small Microcomputer Syst.*, vol. 37, no. 12, pp. 2593–2597, 2016.
- [30] L. Peng, "Cross-domain virtual network mapping algorithm based on minimum cost," *J. South China Univ. Technol. (Natural Sci. Ed.)*, vol. 43, no. 9, pp. 67–73, 2015.
- [31] Z. Zhang, L. Zhu, L. Feng, and L. Liu, "Cross-domain virtual network mapping algorithm based on multidimensional attributes," *J. Univ. Electron. Sci. Technol. China*, vol. 48, no. 2, pp. 174–181, 2019.
- [32] A. Leivadreas, C. Papagianni, and S. Papavassiliou, "Efficient resource mapping framework over networked clouds via iterated local search-based request partitioning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1077–1086, Jun. 2013.
- [33] L. Zhou, "SDN virtual network remapping algorithm design and ONOS distributed database research," *Univ. Electron. Sci. Technol. China*, vol. 2, pp. 19–23, 2017.
- [34] H. Yao, B. Zhang, P. Zhang, S. Wu, C. Jiang, and S. Guo, "RDAM: A reinforcement learning based dynamic attribute matrix representation for virtual network embedding," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: [10.1109/TETC.2018.2871549](https://doi.org/10.1109/TETC.2018.2871549).
- [35] 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.
- [36] M. Chowdhury, F. Samuel, and R. Boutaba, "PolyViNE: Policy-based virtual network embedding across multiple domains," *J. Internet Services Appl.*, vol. 4, no. 1, pp. 1–23, 2013.
- [37] F. E. Zaheer, J. Xiao, and A. R. Boutab, "Multi-provider service negotiation and contracting in network virtualization," in *Proc. IEEE Netw. Oper. Manage. Symp.-NOMS 2010*, 2010, pp. 471–478.
- [38] F. Esposito, I. Matta, and Y. Wang, "VINEA: An architecture for virtual network embedding policy programmability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 11, pp. 3381–3396, Nov. 2016.
- [39] Y. Ni, G. Huang, S. Wu, C. Li, P. Zhang, and H. Yao, "A PSO based multi-domain virtual network embedding approach," *China Commun.*, vol. 16, no. 4, pp. 105–119, 2019.
- [40] D. Wilken and W. Rabbel, "On the application of Particle Swarm Optimization strategies on Scholte-wave inversion," *Geophysical J. Int.*, vol. 190, no. 1, pp. 580–594, 2012.
- [41] N. Kushwaha and M. Pant, "Modified particle swarm optimization for multimodal functions and its application," *J. Huaqiao Univ. (Natural Sci. Ed.)*, vol. 78, no. 17, pp. 23917–23947, 2019.
- [42] P. Zhang, S. Wu, M. Wang, H. Yao, and Y. Liu, "Topology based reliable virtual network embedding from a QoE perspective," *China Commun.*, vol. 15, no. 10, pp. 38–50, Oct. 2018.
- [43] D. Dietrich, A. Rizk, and P. Papadimitriou, "Multi-domain virtual network embedding with limited information disclosure," in *Proc. IFIP Netw. Conf., Brooklyn, NY*, 2013, pp. 1–9.
- [44] Z. Li, X. Zheng, and H. Zhang, "Virtual network mapping algorithm based on multi-objective particle swarm optimization," *Comput. Appl.*, vol. 37, no. 3, pp. 755–759, 2017.
- [45] H. Yao, X. Chen, M. Li, P. Zhang, and L. Wang, "A novel reinforcement learning algorithm for virtual network embedding," *Neurocomputing*, vol. 284, no. 5, pp. 1–9, 2018.
- [46] H. Cao, S. Wu, G. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1406–1416, Feb. 2020.



Peiyang Zhang received the bachelor's and master's degrees from the College of Computer and Science and Technology, China University of Petroleum (East China), in 2003 and 2006, respectively, and the Ph.D. degree from the School of Information and Communication Engineering, the Beijing University of Posts and Telecommunications, in 2019. He is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China). He has authored or coauthored multiple IEEE/ACM Trans./Journal/Magazine papers, such as IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, *ACM Transactions on Asian and Low-Resource Language Information Processing*, IEEE INTERNET OF THINGS JOURNAL, *Computer Communications*, *IEEE Communications Magazine*, and etc. His research interests include semantic computing, future internet architecture, network virtualization, and artificial intelligence for networking.



Chao Wang is a graduate student with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include network artificial intelligence and network virtualization.



Gagangeet Singh Aujla (Senior Member, IEEE) received the B.Tech. and M.Tech. degrees in computer science and engineering from Punjab Technical University, Jalandhar, Punjab, India, in 2003 and 2013, respectively, and the Ph.D. degree in computer science and engineering from the Thapar Institute of Engineering and Technology, Patiala, Punjab, India, in 2018. He is currently working with the Department of Computer Science, Durham University, Durham, U.K. Prior to this, he was working as a Research Associate with Indo-Austria Research Project sponsored

by Department of Science and Technology, Government of India and Ministry of Science, Austria. He was also a Project Fellow with Haryana State Center of Science and Technology funded research project on Smart Grid. He was the recipient of the 2018 IEEE TCSC outstanding Ph.D. Dissertation Award. He is on the Editorial Board of the Sensors Journal. He has guest edited a number of Special Issues in IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE NETWORK, COMPUTER COMMUNICATIONS, SOFTWARE: PRACTICE AND EXPERIENCE, and TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES. He is a Senior Member of the ACM.



Neeraj Kumar (Senior Member, IEEE) received the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra (Jammu and Kashmir), India, in 2009, and was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is working as a Full Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab, India. He is an Adjunct Professor with the King Abdulaziz University, Jeddah, Saudi Arabia and also with the Asia University, Taiwan. He is Visiting Research Fellow

with Coventry University, U.K. He is a Technical Editor of *IEEE Network Magazine*, and *IEEE Communication Magazine*. He is an Associate Editor for the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, *ACM Computing Surveys*, JNCA, Elsevier, IJCS, Wiley, and Security and Communication, Wiley and on the Editorial Board of Computer Communications, Elsevier. He has more than 13 500 citations to his credit with current h-index of 64. He has won the Best Papers Award from IEEE SYSTEMS JOURNAL and ICC 2018, Kansas city in 2018. He has edited more than ten journals special issues of repute and published four books from CRC, Springer, IET U.K., and BPB publications.



Mohsen Guizani (Fellow, IEEE) received the B.S. (with distinction) and M.S. degrees in electrical engineering, and M.S. and Ph.D. degrees in computer engineering from Syracuse University, New York, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor with the Computer Science and Engineering Department, Qatar University. Previously, he served in different academic and administrative positions with the University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri-Kansas City, the University of

Colorado-Boulder, and Syracuse University. He is the author of nine books and more than 500 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the Editor-in-Chief of IEEE Network, serves on the Editorial Boards of several international technical journals, and is the Founder and Editor-in-Chief of the *Wireless Communications and Mobile Computing journal* (Wiley). He has guest edited a number of Special Issues in IEEE journals and magazines. He was a TPC member, Chair, and General Chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He was also the recipient of the 2017 IEEE Communications Society WTC Recognition Award as well as the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and ad hoc sensor networks. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He was a IEEE Computer Society Distinguished Speaker and is currently an IEEE ComSoc Distinguished Lecturer. He is a Senior Member of ACM.