

A Virtual Network Resource Allocation Model Based on Dynamic Resource Pricing

XIAN-CUI XIAO^{1,2}, XIANG-WEI ZHENG^{1,2}, YI WEI^{1,2}, AND XIN-CHUN CUI³

¹School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

²Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250014, China

³School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China

Corresponding author: Xiang-Wei Zheng (xwzhengcn@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61373149.

ABSTRACT The key to network virtualization technology is virtual network mapping, which has been proven to be an NP-hard problem. At present, the methods to solve the problem of virtual network mapping still have the following defects. Most of the existing literature is limited to static virtual network (VN) mapping and static linear resource pricing, which rely on peak allocation and don't meet the user dynamic resource requirements. Therefore, this paper proposes a virtual network resource allocation model based on dynamic resource pricing named GSO-RBFDM. Firstly, group search optimization (GSO) is used to optimize the node mapping scheme during the network mapping process to reduce the cost of network mapping. Secondly, a dynamic nonlinear resource pricing model is established, and genetic algorithm (GA) is used to more accurately search a low-cost network mapping path instead of the traditional Dijkstra algorithm. Finally, virtual network dynamic modeling is performed according to the user dynamic resource requirements, and radial basis function (RBF) is used to predict resource requirements to realize the dynamic resource allocation to users. Simulation results show that, compared with traditional virtual network mapping algorithms, GSO-RBFDM can not only realize dynamic resource allocation, but also show good performance in terms of acceptance rate, network cost, link pressure and average network revenue.

INDEX TERMS Network virtualization, dynamic resource pricing, group search optimization (GSO), genetic algorithm (GA), radial basis function (RBF), dynamic resource allocation.

I. INTRODUCTION

With the continuous development of the availability of new technologies and practical applications, the problem of resource allocation in the emerging decentralized communication environment has become a research focus [1], [2]. As an important resource allocation method, network virtualization technology can customize personalized network services according to users' requirements [3], [4]. However, most existing network mapping methods are static network mapping methods, which rely on peak allocation [5]–[7]. As the users' resource requirements change with the current network load changes in reality, the static resource allocation method will inevitably cause a lot of waste of resources. In view of this, the dynamic network mapping method is proposed to minimize the redundancy of resource allocation without affecting network services [8]–[11]. Specifically, the

dynamic network mapping method calculates the required physical node resources and physical link resources before any virtual network request (VNR) arrives, and then allocates resources according to the user's dynamic resource requirements. However, few literature studies have explored the real-time resource requirements of users to complete dynamic resource allocation. Therefore, this paper attempts to solve the above problems, and finds a large number of literatures shows that RBF algorithm, as a neural network algorithm in machine learning, has good prediction, classification and diagnosis functions. In the dynamic network mapping problem mentioned above, the user's resource requirements change with the current network load. Therefore, it may be a promising method to use RBF algorithm to predict resource requirement and complete the dynamic allocation of network resources [12]–[14].

In the modern competitive economic environment, the use of price as a control mechanism to control network resource allocation is very transparent to users, and the implementation

The associate editor coordinating the review of this manuscript and approving it for publication was Yulei Wu¹.

process is simple and fast. It has been proved that dynamic pricing is better than static pricing in theory [15], and its feasibility has also been proved in practice [16], [17]. Therefore, this paper proposes a dynamic nonlinear resource pricing method to form a sense of resource competition among users. In the traditional network model, the pricing of link resources is static and does not consider the cost of physical nodes on the mapping path. Therefore, it is undoubtedly the quickest and most effective to use the Dijkstra algorithm to find the shortest mapping path.

In this paper, on the one hand, the link and node unit prices have changed over time, which is no longer a simple linear relationship and increases the complexity of the problem. On the other hand, when the mapped path passes through a physical node, the node will allocate a service rate and buffer space for it. Since the network service rate is related to the average delay and the buffer space is related to the packet loss rate, the cost of the node must be considered. The GA algorithm transforms the problem-solving process into a process similar to the crossover and mutation of chromosomal genes in biological evolution. When solving more complex combinatorial optimization problems, it is usually faster to obtain better optimization results than some conventional optimization algorithms. In summary, GA is superior to Dijkstra algorithm in terms of stability and expected effect in solving the shortest path problem in this dynamic weight system.

As far as we know, there is no research on the combination of dynamic resource pricing, intelligent learning algorithm and neural network algorithm to address the dynamic mapping problem of virtual network mapping. The main contributions of this paper are as follows.

(1) GSO is applied to optimize the network mapping scheme to reduce network costs. The individual in each population in GSO is defined as a mapping scheme, in which each population update will produce a producer, 80% of the scroungers and 20% of the rangers [18], [19]. In the network mapping problem of this paper, the network mapping scheme can be optimized to obtain a low-cost mapping scheme with continuous iteration of the population.

(2) A dynamic and nonlinear resource pricing method is proposed, and GA is used to find a low-cost mapping path. Since the resource prices in dynamic nonlinear resource pricing varies with the remaining quantity, the mapping path can be optimized by selection, mating and mutation operators of GA, thereby reducing network costs.

(3) Dynamical resource requirements of users are formulated and supervised RBF is used to predict users' real-time resource requirements to complete dynamic resource allocation. We first establish a dynamic resource requirement model and collect previous data for training. After the user's real-time resource requirement is predicted, then the real-time resource allocation is adjusted accordingly.

It is worth noting that we simply defined a nonlinear model in terms of resource pricing, which may not be suitable for any scenario in reality, so in practical applications, more

factors should be considered to dynamically set resource pricing. In addition, it is well known that intelligent prediction based on neural networks is an effective method to solve nonlinear prediction problems, but its disadvantage is the "black box" nature, which means that you do not know how and why the neural network will get a certain output. Therefore, the predicted results of RBF may not be very accurate and have a certain degree of uncertainty. In view of the above problems, we hope that this dynamic resource pricing and dynamic resource allocation ideas will provide inspiration for future network mapping problems, which urgently need to make more explorations and improvements later.

The rest of this paper is organized as follows. The recent related research works about VN technology are reviewed in Section 2. Section 3 presents the problem description, including network description, dynamic resource requirement model and dynamic resource pricing model; Section 4 describes GSO-RBFD algorithm and Section 5 details the experiments and analysis. Finally, Section 6 concludes the paper and points out the future works.

II. RELATED WORK

As an important technology of information networks, VN allocation technology includes two specific types of resource allocation methods: static resource allocation and dynamic resource allocation. The typical static network allocation method allocates users sufficient resources to meet their peak requirement, ignoring the limited number of resources or other issues [20]–[23]. There are many literatures that used different strategies to improve resource utilization, including sub-graph isomorphism [24], path splitting and migration [25], topology-aware node ranking [26] or other heuristics [27]–[29]. Mosharaf NM, Chowdhury K designed D-ViNE and R-ViNE mapping algorithms by using deterministic and random rounding techniques respectively [30]. Wang and Hamdi formulated the VN embedding problem as a new multiple objective linear programming optimization program, and proposed an efficient online heuristic VN embedding algorithm Presto based on Blocking Island (BI) to maximize the revenue [31]. He and Zhuang *et al.* proposed to apply spectral clustering based on field theory to extract substrate network features and established a dynamic region of interest to find embedding areas with energy-saving potential for virtual networks [32]. Wang and Bi *et al.* modeled the VNE problem as Markov Decision Process (MDP) and developed a neural network to approximate the value function of VNE states [33]. Zhao and Parhami proposed to use graph eigenspace notions for node mapping and contributed an inexact algorithm which projects all nodes to a 2-D eigenspace to generate node mapping schemes [34]. Yan and Ge *et al.* combined reinforcement learning with a novel neural network structure and proposed a new virtual network embedding algorithm to complete virtual network mapping automatically [35]. Yuan and Tian *et al.* proposed a VNE algorithm for data center topology based on the Q-learning algorithm, in which an agent for each VN designed a reward

function related to the effect of virtual link embedding [36]. In addition, some other studies have shown that intelligence optimization algorithm is also an effective way to solve the network mapping problem. Li and Yang *et al.* proposed an optimization study of GA, including seed selection, mating and mutation for solving network resource deployment [37]. Mijumbi, Gorricho and Serrat proposed a study using various optimization results to avoid the singularity of the solution of the differential evolution algorithm [38]. Yu, Yi and Rexford used a redefined particle swarm optimization algorithm to solve the problem of VN mapping which show superiority to some extent [39].

Many existing literatures have analyzed and explored the allocation of resources and management issues. However, how to reduce network redundancy through dynamic resource allocation has not been well solved. Dynamic network mapping is a way to study real-time resource allocation problems. This method can dynamically calculate the physical node resources and physical link resources required for the virtual request before the VNR arrives, and then allocate the node and link resources according to the user's real-time resource requirements [40]. Li, Zou and Wei proposed a new virtual network embedding algorithm to improve the original subgraph isomorphism search process and overcome the defects of existing virtual network embedding algorithms [24]. Fickas and Feathe used the PageRank algorithm to deal with network mapping problems to increase acceptance rates and revenues [41]. Oveis and Amjady described a dynamic monitoring method to analyze the impact of related environmental changes on performance [42]. Although the previous work was to optimize the network mapping environment to reduce resource waste, we will have different work in this paper. Our research focuses on using RBF to predict user requirement and reduce resource allocation redundancy, as well as dynamic pricing of resources and the use of GA to optimize the mapping path. Moreover, as a feed-forward neural network algorithm with good performance, although RBF algorithm has become a widely accepted tool in in many economic and management model fields [43]–[45], there are few studies on the application of RBF algorithm in dynamic network resource allocation. In the dynamic network mapping, the user's resource requirements change dynamically over time. Therefore, using the RBF algorithm to predict user resource requirements and complete resource dynamic allocation can reduce resource allocation redundancy and improve resource utilization.

In view of the above research and application of intelligent algorithms and neural network algorithms and the research of VN mapping algorithm, we propose a virtual network resource allocation model based on dynamic resource pricing named GSO-RBFDM. It should be noted that our work is different from the above research. This is the first time that an intelligent learning algorithm and a neural network algorithm have been combined to study the VN dynamic resource allocation. Firstly, we use GSO to optimize the network mapping scheme and complete the initial allocation. Secondly, the

supervised RBF is used to predict the user's node resource and link resource requirements, and then we adjust the allocation of network resources based on the prediction results. Finally, we establish a dynamic nonlinear resource pricing model, and use GA instead of Dijkstra algorithm to optimize the mapping path to find the low-cost network mapping path more accurately.

III. NETWORK MODEL AND PROBLEM DESCRIPTION

A. NETWORK MODEL

Physical network is normally defined as a weighted undirected graph $G_s = (N_s, L_s, C_s^n, C_s^l)$, where represents a set of the physical network nodes. The properties of a node include its computing capability, memory, disk capacity, and cache size, network I/O, forwarding delay, processing delay and so on. represents a set of the physical network links. The properties of a link include the amount of bandwidth, transmission delay and bit error rate. It is unrealistic to fully consider all the factors, and therefore, the node attributes in this paper include the computing capability and geographic location while the link attribute only includes bandwidth. represents the computing capability of each node and represents the physical link bandwidth.

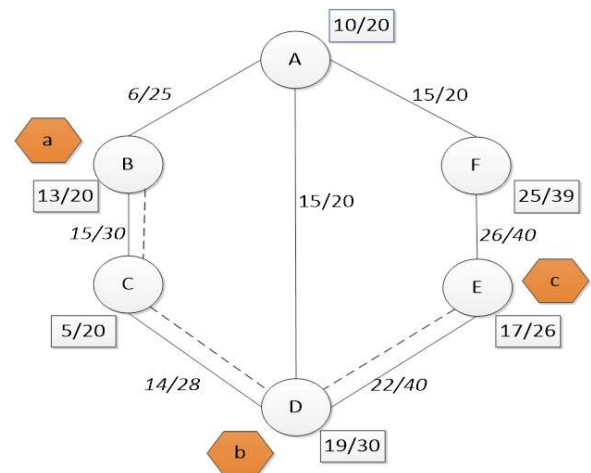


FIGURE 1. Physical network.

Similarly, a VNR can be defined as a weighted undirected graph $G_v = (N_v, L_v, R_v^n, R_v^l)$, where represents a set of VNR nodes, represents a set of VNR links, is the computing capacity requirement of the nodes and is the bandwidth requirements of the virtual links. When a VNR arrives, the physical network allocates node and link resources to satisfy the VNR. If the physical network can't provide the qualified node and link resources, the VNR request will be rejected. Figure 1 shows an example of a physical network, where the numbers in the rectangular boxes represent the available and the total computing resources of each node, while the numbers around the links represent the available and the total bandwidth of the link. Figure 2 shows an example of a VN with 3 nodes and 3 virtual links. Similarly, the numbers in

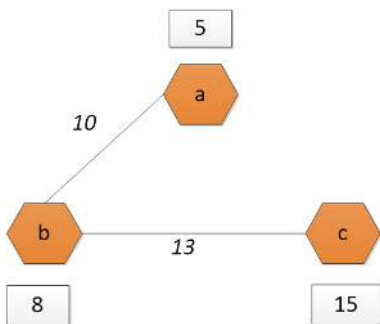


FIGURE 2. Virtual network request.

the rectangle and around the links have the same meaning as Figure 1.

The problem of virtual network mapping in this paper is to solve the virtual request in Figure 2 to find the nodes and links that meet the nodes and link constraints on the physical network in Figure 1 to complete the virtual network mapping process.

B. DYNAMIC RESOURCE REQUIREMENT MODEL

As the number of Internet users continues to rise, the way that network users consume network resources is also changed a lot. In order to construct the user’s dynamic network resource requirement model, we refer to the recent research report of network usage as shown in Figures 3.

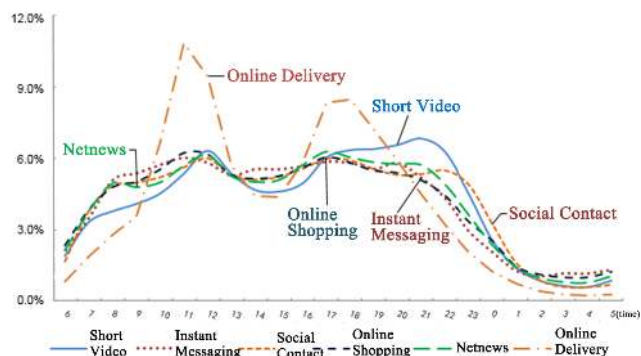


FIGURE 3. Time distribution of user’s mobile phone access to the Internet [46].

Figure 3 shows the time comparison of six types of applications accessing the Internet from the “45th Statistical Report on China’s Internet Development in 2020-Internet Access Environment”. It can be seen that the time distribution curves used by the six types of applications are relatively close, and the peaks of use all start between 8:00 and 10:00 and end between 21:00 and 22:00. During the period, the length of time occupied by Internet resources is relatively evenly distributed, accounting for about 5% to 6%; Short video apps have a peak usage period between 17:00 and 22:00, while the use characteristics of online takeaway apps are more obvious, and their peak periods appear between 11:00 and 12:00 and 17:00-19:00, and the usage time accounts for 20.5% and 24.3%, respectively.

In view of the above investigation, we maybe conclude that users’ requirements for mobile network resources shows a regular peak and valley, which actually forms a regular pattern of long-term resource requests. Therefore, based on the above characteristics of Internet user data statistics, we constructed a dynamic resource requirement model as shown in Figure 4.

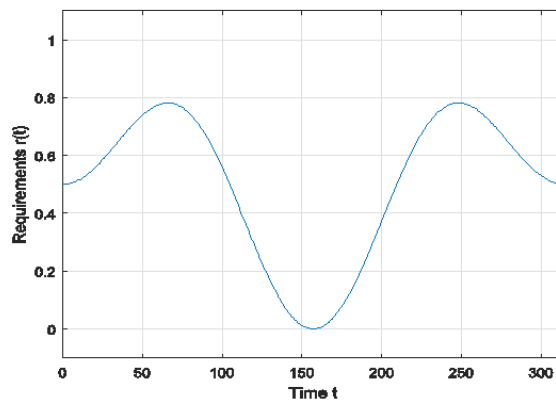


FIGURE 4. Dynamic model of user resource requirement.

Since the use of the Internet in modern society is very complicated, we cannot fully display it for our research. Therefore, based on the above statistical report, we propose a dynamic resource requirement model for resource dynamic requirements of VN nodes and links. Figure 4 shows the user’s dynamic resource requirement trend, where periodic peaks and valleys appear at different points in time, representing the peaks and valleys of the number of Internet users in reality, respectively.

C. DYNAMIC RESOURCE PRICING MODEL

Although the capacity of network links is kept enough with the continuous development of technology, the oversupply of the Internet is still considered to be “Economic luxury” [47]. In a highly competitive economic environment, resource pricing can be used as a control mechanism for allocating network resources. Since traditional static resource pricing is based on a linear function, it does not lead to competition between VN users and maximum profit of infrastructure providers. In traditional static symmetric resource pricing, a long-term bandwidth provisioning agreement is signed between the user and the network provider. This agreement does not guarantee the maximum value of resources in the process of resource supply and requirements, and is gradually replaced by short-term customization. In order to overcome the above shortcomings and maximize the value of resources, we propose a dynamic resource asymmetric pricing method, in which the price of unit resources varies with the percentage of resource usage rather than fixed resource pricing throughout the network supply and requirements. Figure 5 shows a pricing comparison between dynamic asymmetry (D-asymmetry) and static symmetry (S-symmetry).

As shown in Figure 5, S-symmetrical does not give rise to competition when users rent resources, which easily leads to

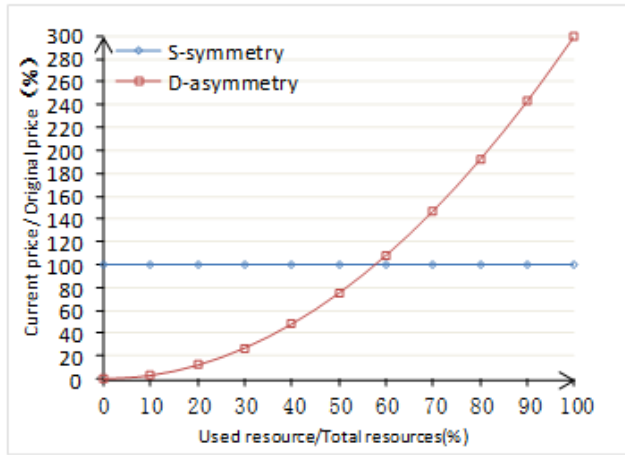


FIGURE 5. Dynamic model of user resource requirement.

overuse of some resources and reduces network acceptance. While in the pricing of D-asymmetry, the unit price increases as the percentage of resource occupancy increases. As the resource pricing is an important factor in network mapping, nodes and links with more remaining resources will be preferred when the unit price of some resources is high. For network users, different physical resources have different unit pricing, and form a competition mode which depends on the ratio between users' cost and revenue.

IV. GSO-RBFD ALGORITHM

A. GSO ALGORITHM

GSO was originally used to solve the multi-mode problem in the continuous domain. However, the optimization of the node scheme in this paper is no longer a continuous threshold problem. Therefore, GSO should be redefined to solve discrete optimization problems. The individual members and associated operations should be redefined in accordance with the specific problems [48]. The specific optimization process in this paper is shown in the Algorithm 1 below.

For any optimization algorithm, the formulation of the objective function determines the purpose of our optimization. Similarly, each individual in GSO refers to a feasible mapping scheme, which includes node mapping and link mapping. Moreover, when calculating the individual fitness value during the GSO iteration process, both the node and link costs are taken into account, and the final mapping scheme is jointly determined by the node mapping and link mapping. The resource pricing proposed in this paper is a dynamic asymmetric pricing form as shown in Figure 5 and we define the objective function of the GSO algorithm including node and link cost as formula (1):

$$f(n^v, l^v)_{\min} = \omega_1 \sum_{n^v \in N^v} (N(n^v) \times pc_n) + \omega_2 \sum_{l^v \in L^v} \left(\sum_{hop=0}^{Hops} (L(l^v) \times pc_b) \right) \quad (1)$$

Algorithm 1 Group Search Optimization Algorithm

Require:

Several feasible initial network mapping schemes, virtual request events with node and link constraints and physical network.

Ensure:

The network mapping scheme of VN on physical network.

- 1: Set group size N, random initialization group and angle;
- 2: Set the maximum number of iterations MG;
- 3: Treat any feasible initial network mapping scheme as an individual;
- 4: The objective function is used to calculate the fitness value of each individual;
- 5: Save the current and the best individual fitness values;
- 6: **while** (The termination condition is not satisfied) **do**
- 7: **for** $i = 0$ to (Number of individuals-1) **do**
- 8: Select a best individual as the producer and then search the space according to the behavior of producer;
- 9: 80% percent of the members will be randomly selected as scroungers, they will be updated according to the scrounger's behavior to get;
- 10: The remaining individual 20% percent of the members will be rangers, they will be updated according to the ranger's behavior to get;
- 11: **end for**
- 12: Update the substrate network resources.
- 13: Calculate the current fitness of each individual value with formula (1);
- 14: Record the current fitness value of each individual;
- 15: Select the best individual as a new producer and update the other individuals;
- 16: **end while**

where $N(n^v)$ represents a collection of physical nodes where virtual nodes are mapped on while $L(l^v)$ represents a collection of underlying network links the virtual links are mapped on; pc_n and pc_b represent the current unit pricing of physical node resources and link resources respectively; and ω_1 and ω_2 are balance parameters between the node cost and link cost.

B. NETWORK PATH MAPPING

Since resources and costs are linear and the cost of nodes on the path is ignored in the traditional network model, most algorithms use Dijkstra's algorithm to find the shortest mapping path. However, the price of resources in this paper is dynamic, and considering that when the mapping path passes through a physical node, the node will allocate a service rate and buffer space for it, so the cost of the node should be considered. Therefore, Dijkstra's algorithm is no longer an effective algorithm for finding low-cost paths in the network mapping problem of this paper. Figure 6 shows an example

where Dijkstra’s algorithm cannot find a low-cost network mapping path.

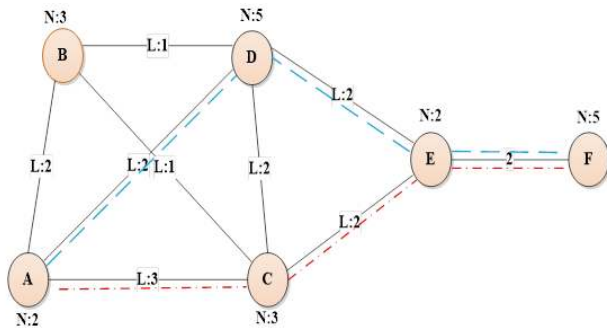


FIGURE 6. Feasible physical mapping paths.

When the mapping path passes through a physical node, the node will allocate a service rate and buffer space for it. The network service rate is related to the average delay, and the buffer space is associated with the packet loss rate. Therefore, in order to find a more reasonable physical mapping path in the path optimization process, the cost of the node must be considered, and the influence of the two on the selection of the mapping path needs to be balanced by corresponding parameters. Therefore, the smaller the sum of the nodes and links passed, the lower the cost is in the path selection process. As shown in Figure 6, the Dijkstra algorithm calculates the shortest path by the hop-by-hop method, the shortest path between the source node A and the destination node F is given by ADEF, where the link cost is $2+2+2=6$, and the node cost is $2+5+2+5=14$. In addition, the shortest path ACEF between the source node A and destination node F with the same hop count has a link cost of $3+2+2=7$ and a node cost is $2+3+2+5=12$. Therefore, in this example, the Dijkstra algorithm cannot find the low-cost shortest path.

For a given network, link cost, node cost, path cost function and source and destination nodes, we hope to find a way to minimize path cost. Therefore, in this paper, our goal is to develop a GA model to find the best way to minimize the path cost. First, we code the routing path, where the sequence in the string represents the order of nodes in the routing path, which is called chromosome in genetic algorithm. The first site of the chromosome is the source node, while the last site of the chromosome is the destination node of the path. Chromosome lengths are less than or equal to the total number of the largest nodes we have set up beforehand. The goal of this paper is to find a low-cost path from the source node to the destination node, including link cost and node cost on the path. In GA algorithm, current chromosomes are evolved by single-point crossover to obtain chromosomes with better fitness. Figure 7 shows an example of single-point cross evolution of a chromosome. In addition, we define the fitness function for each chromosome as shown in formula (2).

$$f(n^v, l^v)_{\min} = \alpha \sum_{l^v \in P(n_s, n_t)} \text{Cost}(l^v) + \beta \sum_{n^v \in P(n_s, n_t)} \text{cost}(n^v) \quad (2)$$

where $P(n_s, n_t)$ represents the path set between initial node A and destination node B, $L(P(n_s, n_t))$ represents the collection of links contained in the path $P(n_s, n_t)$, $N(P(n_s, n_t))$ represents the collection of nodes contained in the path $P(n_s, n_t)$, and α and β are balance parameters between the node cost and link cost.

The parental chromosomes were randomly selected for mating and the probability of chromosome selection is proportional to the fitness of the chromosome. Check the public nodes of both parents, where the parents’ paths intersect, and then randomly select a common node for cross-evolution. Such as, in Figure 7, the common nodes of the parent chromosome are D and F, and we obtain the child chromosome by exchanging the substrings of the parent chromosome. As shown in Figure 7, a single point exchange is performed at the common node D, F in parental chromosome. Then, we exchange substrings DEFHG and DFG to generate sub-chromosomes OP1 and OP2, and exchange substrings FHG and FG to generate OP3 and OP4, respectively.

It is worth noting that, for the simplicity of the path finding process in this paper, if the parent chromosome does not have a common node, no crossover will occur and the child’s chromosome is an exact copy of the parent. Moreover, the child’s chromosome may not be feasible due to node duplication or path disconnection. To solve this, we first remove the duplicate nodes from the child’s chromosomes and then check the child’s path connections and we don’t fix children who are not feasible in our paper. A sub-chromosome is not feasible if two adjacent nodes of a sub-chromosome are not directly connected by a link. If the child is not feasible after the crossover, we will discard the infeasible child and use an exact copy of the parent as a new child.

C. SUPERVISED RBF

RBF is a three-layer feed forward neural network. The first layer, which called the input layer, consists of signal nodes to transmit information. The second layer is the hidden layer of RBF neurons, in which an activation function is set up to solve the linear inseparability problem of low-dimensional linear separation in high-dimensional space. The last layer, output layer is used to adjust the weight of the network through a linear optimization strategy. Figure 8 shows a specific network structure.

Among, $F(x) = \sum_{i=1}^m w_i \varphi(\|x - c_i\|)$ and we obtain the width, the center of the training neuron and the associated parameters of the supervised RBF through a specific prediction method. The relevant parameters value as follow, initial centers $\phi_i = 10$, the value of width σ is between 0.1 and 0.3, width adjustment $\mu = 0.6$, Learning rate $\eta = 0.001$, target error $\xi = 0.5$, iteration number $K = 2000$ and initial weights=1. Figure 9 shows the specific design process of the resource prediction model in this paper.

In this paper, before a new VN event arrives, we pre-collect the resource requirements of multiple users in a cycle time as the training sample set of RBF units, and obtain

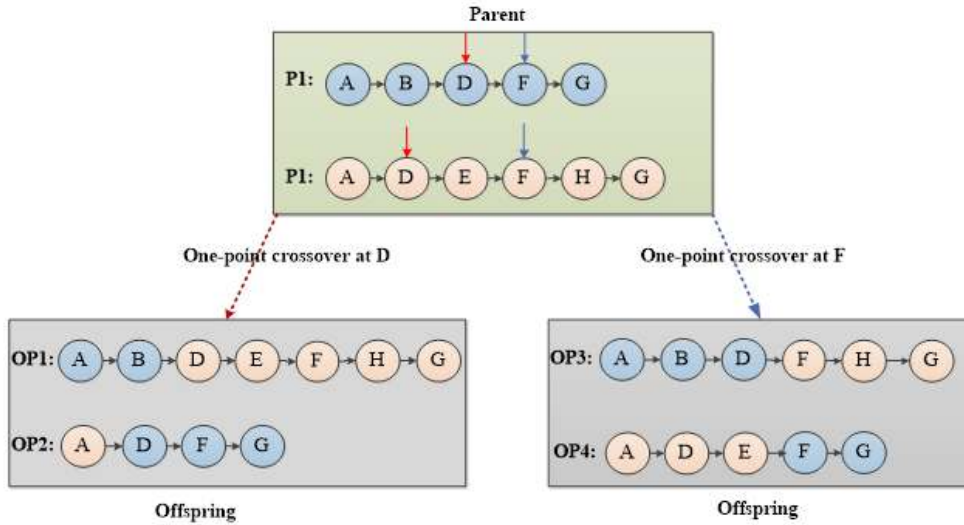


FIGURE 7. New physical mapping path generated by GA.

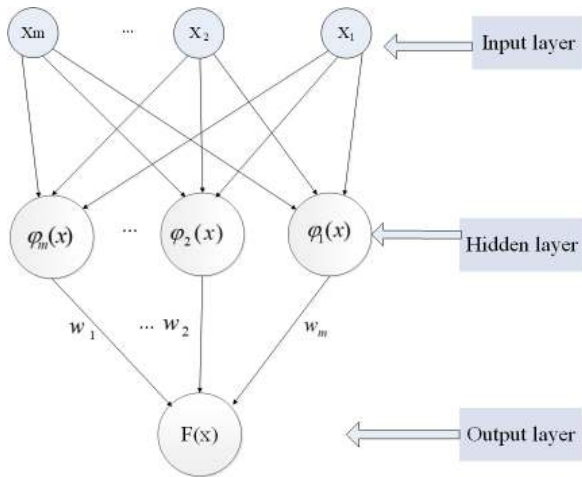


FIGURE 8. RBF neural network structure.

the corresponding neural network structure through multiple trainings. When a new VNR arrives, the GSO is used to optimize the mapping scheme to complete the initial allocation, and then the trained RBF is used to predict the resource requirement of the user, and finally reallocate the physical resources according to the predicted result. We use functions to model the volatility requirements of VNR resources and manage the physical resources at each moment throughout the lifecycle.

D. DESCRIPTION OF GSO-RBFDM

In order to better express the association of each part of this paper, including dynamic network mapping, dynamic pricing and the application of genetic algorithm, we give the flowchart of GSO-RBFDM algorithm as shown in Figure 10.

E. PSEUDO CODE OF GSO-RBFDM

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL SETTINGS

In this paper, the topologies of the physical networks and VNRs are randomly generated by GT-ITM tool [49]. For any VNR request event, the computing resource requirements of each virtual node and bandwidth requirements of each virtual link are randomly generated. Table 1 shows the parameters of the simulation experiment in this paper.

TABLE 1. Simulation parameters.

Experimental variables	Value
Node number of the physical network	100
Link number of the physical network	950
Available computing resources on physical nodes	50-100units
Available bandwidth resources on physical links	50-100units
Bandwidth requirement of a virtual link	0-50units
Computing resource requirement of a virtual node	0-50units
Node number in a VNR	10-25
Connection probability between substrate nodes	0.5

B. EVALUATION INDICATORS

The main evaluation index of the network is defined as follows:

1) Network acceptance rate:

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T VNR}{\sum_{t=0}^T VNR_s} \tag{3}$$

where $\sum_{t=0}^T VNR$ represents the number of virtual networks successfully mapped from time $t = 0$ to time T and $\sum_{t=0}^T VNR_s$

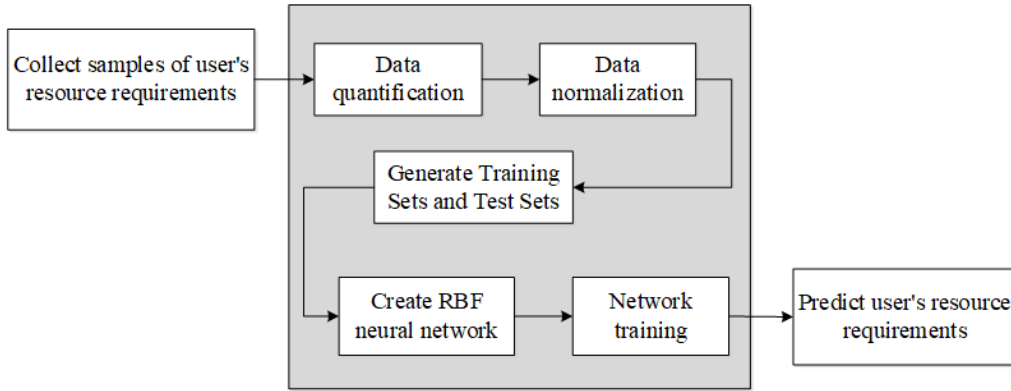


FIGURE 9. Dynamic resource requirement prediction process.

TABLE 2. Simulation parameters.

Algorithm name	Resource Supply	Path optimization method	Mapping method	Resource allocation
GSO-RBFDM	Dynamic	GA	GSO	Reallocate resource based on RBF predicts
GSO-STACND	Static	GA	GSO	Fixed throughout the lifecycle
RANDOM-RBFDM	Dynamic	GA	Random	Reallocate resource based on RBF predicts
RANDOM-STACND	Static	GA	Random	Fixed throughout the lifecycle
GREEDY-RBFDM	Dynamic	GA	Greedy	Reallocate resource based on RBF predicts
GREEDY-STACND	Static	GA	Greedy	Fixed throughout the lifecycle

represents the total number of virtual network requests from the time $t = 0$ to the time T .

2) Average network cost:

$$\lim_{T \rightarrow \infty} \sum_{t=0}^T \text{Cost}(G_v, t) / \text{VNR}_s \quad (4)$$

where $\sum_{t=0}^T \text{Cost}(G_v, t)$ represents the resource required to successfully map a virtual network from $t = 0$ to T . 3) Node pressure:

$$\text{Node_Load}(n^s) = \sum_{v \in E^v} \text{Map}_{n^s}(C_n^v) / C_n^s \quad (5)$$

where C_n^s represents the node resource of physical node, $\text{Map}_{n^s}(C_n^v)$ represents the sum of the resources of all the virtual nodes mapped on.

3) Link pressure:

$$\text{Link_Load}(l^s) = \sum_{v \in E^v, p \in (p, v)} \text{Map}_{l^s} C_l^v(p, v) / C_l^s \quad (6)$$

where C_l^s represents the physical link, $\text{Map}_{l^s} C_l^v(p, v)$ represents the sum of resources used.

4) Average network revenue:

$$\lim_{T \rightarrow \infty} \sum_{t=0}^T \text{Rev}(G_v, t) / T \quad (7)$$

where $\sum_{t=0}^T \text{Rev}(G_v, t)$ represents the total income during time $t=0$ to T as the virtual network are successfully mapped on physical network.

C. ANALYSIS

In this section, we will evaluate the advantages of the GSO-RBFDM proposed in our paper. GSO-RBFDM is a dynamic virtual network mapping algorithm that combines GSO and RBF. Firstly, GSO is used to optimize the network mapping scheme and allocate the initial resource for the virtual request. Then, as user resource requirements change, a supervised RBF neural network is used to predict user's resource requirements. Finally, reallocate network resources for users according to the predicted result and update the physical resources. To explain the performance of GSO-RBFDM, the four comparison algorithms in the paper are briefly described as in Table 2.

We improve the traditional greedy algorithm and random algorithm, such as applying genetic algorithm to link mapping and combining RBF algorithm to complete network mapping, so as to fit the simulation environment and ensure the fairness of different algorithms in the simulation experiment process. In addition, several evaluation indexes were selected in the comparison simulation experiment, including acceptance rate, average cost, link pressure and average revenue. The experimental results are shown in Figure 11- Figure 14.

As shown in Figure 11, GSO-RBFDM has the highest acceptance rate, followed by GSO-STAC, while the acceptance rates of RANDOM-RBFDM and GREEDY-RBFDM are higher than those of RANDOM-STAC and GREEDY-STAC, respectively. Generally, the acceptance rate of random algorithm and greedy algorithm will be different under different network resource settings and topologies. Moreover, it

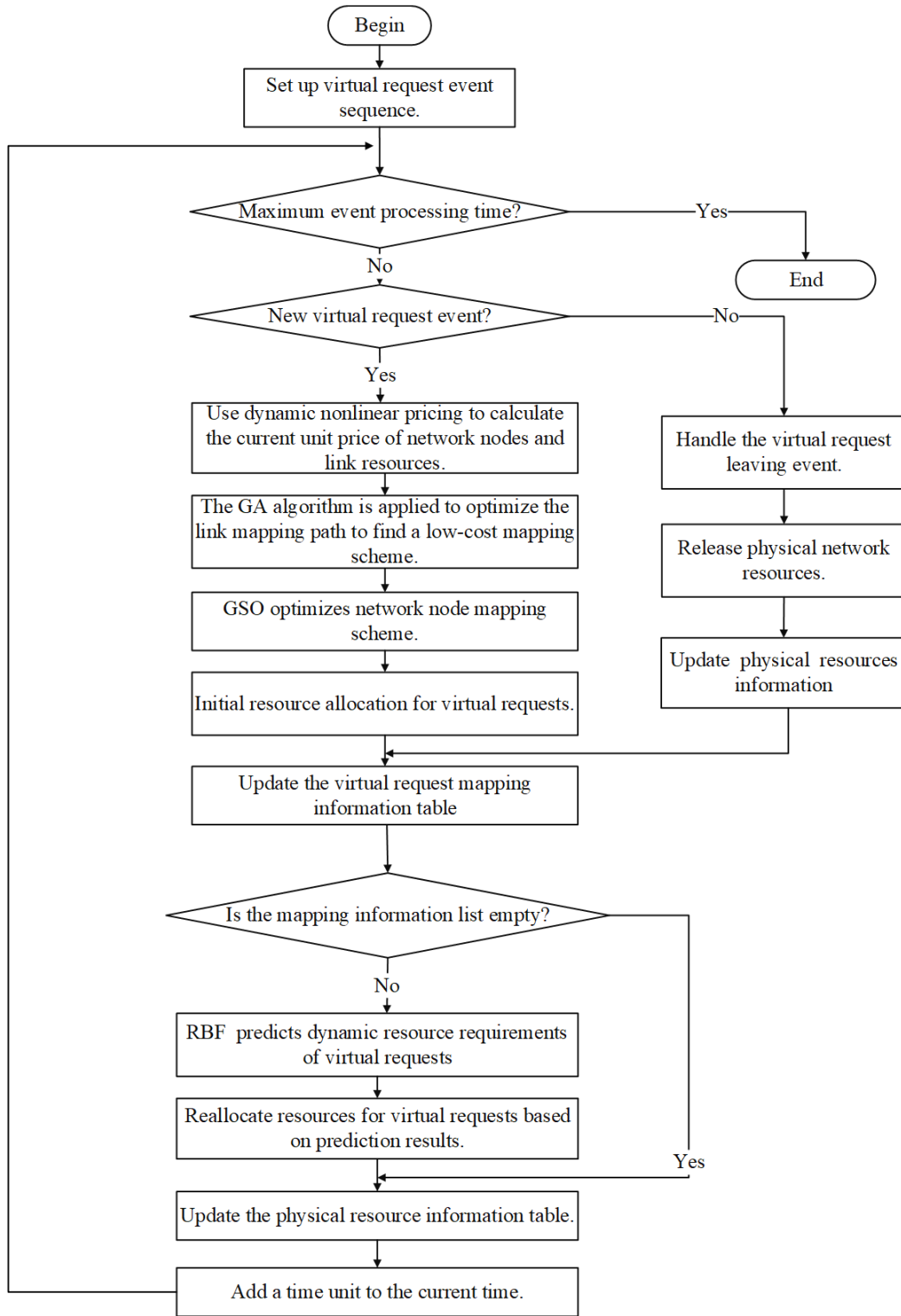


FIGURE 10. GSO-RBFD flow chart.

can be seen that compared with the static mapping method, the dynamic mapping method has a higher acceptance rate. Because in the dynamic network mapping algorithm, RBF is used to predict the resource requirements of virtual nodes

and links, and then reallocate resources according to the prediction results, thereby reducing the redundancy of resource allocation and saving network resources to a certain extent. For this reason, compared with static network mapping

Algorithm 2 Dynamic Mapping of GSO-RBFDM

Require:
 virtual request events with node and link constraints and physical network;

Ensure:
 Event processing results and resource recycling results;

- 1: Generate a physical network and a series of virtual request events.
- 2: The start time of the event is the arrival time of the first event.
- 3: Save all the request termination time and the maximum time in the last event;
- 4: **for** $i = 0$ to (the Max moment of last event - 1) **do**
- 5: **if** (A new virtual network request arrives) **then**
- 6: **if** (Current moment== Request arrival time) **then**
- 7: Take this VN from the event.
- 8: **if** (Satisfy constraints) **then**
- 9: Generate feasible initial network mapping schemes X_i ;
- 10: Use Algorithm 1 to find a better network mapping scheme;
- 11: Take X_i as the final node mapping scheme;
- 12: **else**
- 13: Reject this request.
- 14: **end if**
- 15: **else**
- 16: Record the actual requirements as input data.
- 17: **if** (Satisfy training needs) **then**
- 18: Use resource prediction model to predict resource requirements of users;
- 19: Allocate the actual resource requirements according to the prediction result;
- 20: **else**
- 21: Record the actual resource requirements and allocate resources for this request according to the last allocation scheme;
- 22: **end if**
- 23: **end if**
- 24: Release allocation resources at the last moment;
- 25: **else**
- 26: The network mapping process is completed;
- 27: **end if**
- 28: Turn to the next moment.
- 29: **end for**

algorithms, dynamic mapping algorithms have relatively sufficient network resources to provide services for more virtual network requests. In addition, the GSO algorithm selects a better fitness function from a variety of possible mapping schemes as the final mapping scheme, which can achieve more reasonable network mapping than the greedy algorithm and the random algorithm. Therefore, the acceptance rate of GSO-RBFDM and GSO-STAC is higher than other comparison algorithms.

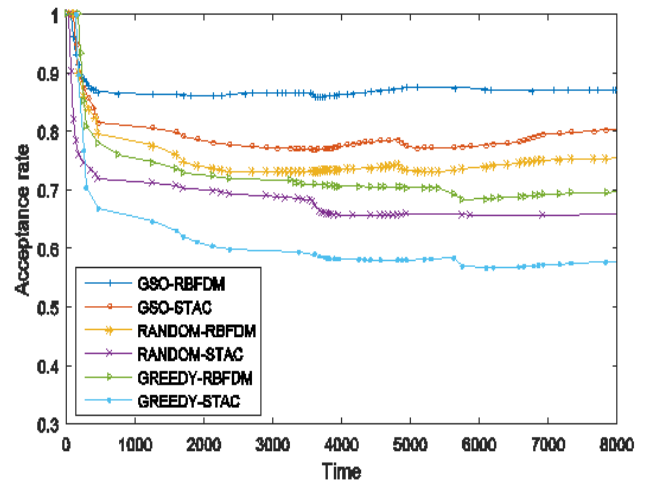


FIGURE 11. Acceptance rate.

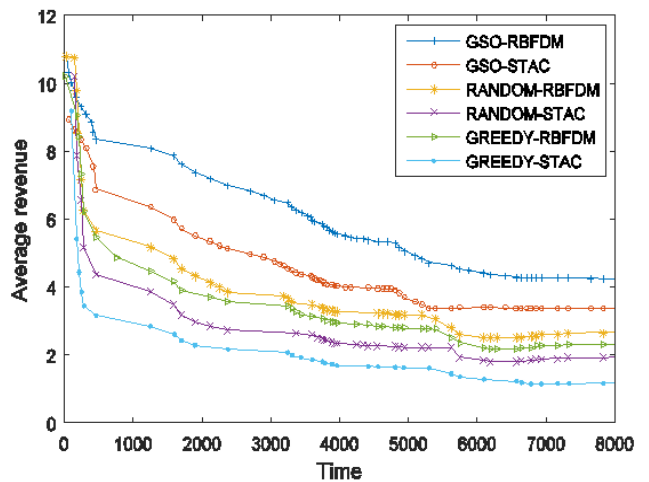


FIGURE 12. Average cost.

As shown in Figure 12, compared with the static mapping algorithms GSO-STAC, RANDOM-STAC and GREEDY-STAC, the dynamic mapping algorithms GSO-RBFDM, RANDOM-RBFDM and GREEDY-RBFDM have lower costs respectively. Because in the static network mapping algorithm, a fixed peak resource is allocated in order to cope with the occasional maximum network load of users. In the dynamic network mapping algorithm, the user's resource requirement is allocated based on the RBF prediction result, and the prediction result is usually lower than the user's fixed peak requirement due to dynamically changing network load. Therefore, the resource occupancy rate of the dynamic network mapping algorithm is lower than that of the static network mapping algorithm, which results in lower network costs. In addition, compared with RANDOM-STAC, RANDOM-RBFDM, GREEDY-RBFDM and GREEDY-STAC, the average network cost of GSO-RBFDM and GSO-STAC is lower. We know that the average network cost includes node cost and link cost. When a new virtual virtual request arrives, the GSO algorithm updates

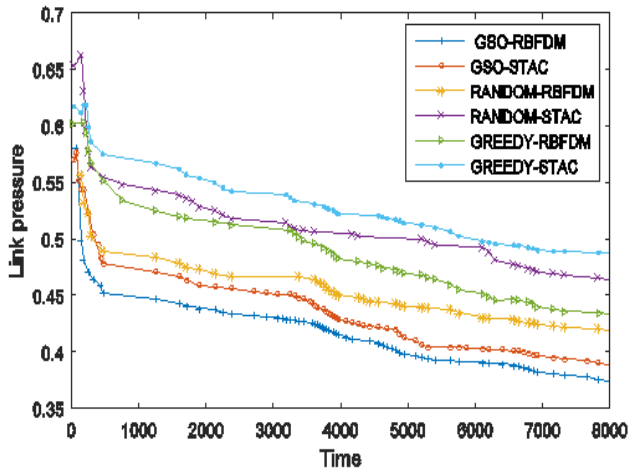


FIGURE 13. Link pressure.

each individual (i.e. network mapping scheme) through the transformation between the producer, the scrounger and the ranger, which is different from the one-time adaptation network of the greedy algorithm and the random algorithm Mapping. At the same time, the GA algorithm is used to optimize the physical mapping path in multiple individual iterations of GSO, which reduces the cost of link mapping and thereby reduces the average network cost.

As shown in Figure 13, the link pressures of GSO-RBFDM, RANDO-RBFDM and GREEDY-RBFDM are lower than those of GSO-STAC, RANDOM-STAC and GREEDY-STAC, respectively. Because the static network mapping algorithm allocates network resources according to the user’s original link resource requirements, and no longer adjusts the resource allocation when the user’s resource requirements change due to network load. In the dynamic network mapping algorithm, RBF is used to predict the resource requirements of virtual requests and dynamically adjust resource allocation for users after the initial resource allocation is completed. The dynamic resource allocation method essentially saves some network resources and meets the user’s basic resource requirements. Therefore, the dynamic network mapping algorithm has higher link resource utilization and lower link cost than static network mapping. In addition, the link pressure is affected by the acceptance rate and the mapping method. The link pressure of GREEDY-RBFDM is higher than RANDOM-RBFDM and lower than RANDOM-STAC. As GSO-RBFDM and GSO-STACND use GSO to optimize network mapping schemes to improve resource utilization, the link pressures are lower than the other algorithms respectively. It is worth noting that in the process of running the RANDOM-STAC algorithm, there are fewer events that happen to be processed in the time period of 5000-6000, so the average link pressure trend is smoother than other algorithms, which is normal and has a certain chance.

As shown in Figure 14, GSO-RBFDM and GSO-STACND have the highest average revenue, followed by RANDOM-RBFDM and GREEDY-RBFDM, RANDOM-STAC, and

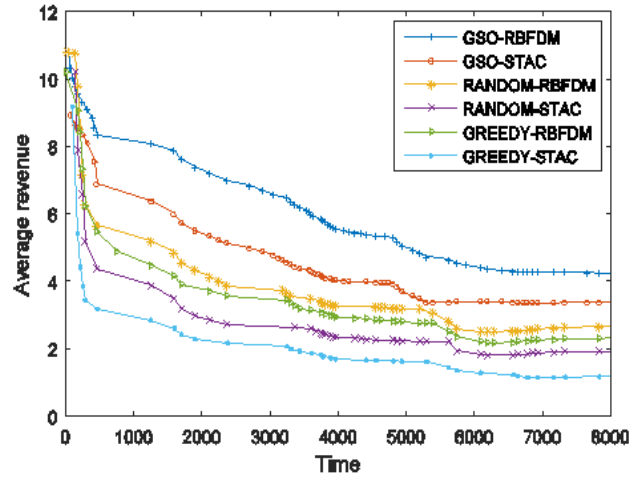


FIGURE 14. Average revenue.

GREEDY-STAC has the lowest average revenue. Since the network revenue comes from the network fees paid by the network users, the network average revenue comparison results are similar to the acceptance rates in Figure 11. In the same physical network environment, the more virtual requests are accepted, the higher the network average revenue will be. In general, based on the above analysis of the cost, node pressure and link pressure factors of the algorithm, the GSO-RBFDM proposed in this paper not only satisfies the user’s dynamic resource requirements, but also improves the network utilization and average revenue. These results also reflect the effectiveness of the proposed algorithm GSO-RBFDM.

VI. CONCLUSION AND FUTURE WORK

By summarizing the limitations of existing static resource pricing and the defects of static resource allocation, this paper proposes a virtual network resource allocation model based on dynamic resource pricing named GSO-RBFDM. In the network mapping, the GSO is used to optimize the node mapping scheme and supervise the RBF to predict the users’ resource requirements. GSO-RBFDM can realize dynamic resource allocation and improve resource utilization efficiency by adjusting the allocated network resources. Simulation results show that, compared with the static network mapping algorithm, the dynamic network mapping algorithm shows good performance in terms of acceptance rate, average cost, link pressure and average revenue.

Further work includes the following three aspects: 1) Study the adaptability of GSO-RBFDM in different physical network environments and the impact of the number of resources on the performance of the algorithm. 2) Optimize the resource requirement training data set to improve the prediction accuracy of node and link resource requirements and reduce the network cost.

REFERENCES

[1] Z. F. Xiao and Y. Xiao, “Security and privacy in cloud computing,” *IEEE Commun. Surveys Tuts.*, vol. 2, no. 2, pp. 843–859, Jan. 2013.

- [2] C.-T. Yang, S.-T. Chen, J.-C. Liu, Y.-Y. Yang, K. Mitra, and R. Ranjan, "Implementation of a real-time network traffic monitoring service with network functions virtualization," *Future Gener. Comput. Syst.*, vol. 93, pp. 687–701, Apr. 2019.
- [3] M. Chowdhury, F. Samuel, and R. Boutaba, "Poly-ViNE: Policy-based virtual network embedding across multiple domains," in *Proc. 2nd ACM SIGCOMM Workshop Virtualized Infrastruct. Syst. Archit. ACM*, 2010, pp. 49–56.
- [4] L. Wenzhi, L. Shuai, X. Yang, and T. Xiongyan, "Dynamically adaptive bandwidth allocation in network virtualization environment," *Int. J. Adv. Inf. Sci. Service Sci.*, vol. 4, no. 1, pp. 10–18, Jan. 2012.
- [5] 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.
- [6] F. Zhu and H. Wang, "A modified ACO algorithm for virtual network embedding based on graph decomposition," *Comput. Commun.*, vol. 80, pp. 1–15, Apr. 2016.
- [7] M. Dorigo and S. Thomas, "Ant colony optimization: Overview and recent advances," *Comput. Commun.*, vol. 146, pp. 227–263, Sep. 2010.
- [8] D. Minarolli and B. Freisleben, "Utility-driven allocation of multiple types of resources to virtual machines in clouds," in *Proc. IEEE 13th Conf. Commerce Enterprise Comput.*, Sep. 2011, pp. 137–144.
- [9] S. Zhang, Z. Qian, J. Wu, S. Lu, and L. Epstein, "Virtual network embedding with opportunistic resource sharing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 816–827, Mar. 2014.
- [10] S. Zhang, Z. Qian, J. Wu, and S. Lu, "An opportunistic resource sharing and topology-aware mapping framework for virtual networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2408–2416.
- [11] G. Sun, H. Yu, L. Li, V. Anand, Y. Cai, and H. Di, "Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter," *Photonic Netw. Commun.*, vol. 23, no. 2, pp. 109–122, Apr. 2012.
- [12] D. Sun, G. Chang, L. Sun, and X. Wang, "Surveying and analyzing security, privacy and trust issues in cloud computing environments," *Procedia Eng.*, vol. 15, pp. 2852–2856, 2011.
- [13] K. Benzekki, A. El Fergougui, and A. El Alaoui, "A secure cloud computing architecture using homomorphic encryption," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 293–298, 2016.
- [14] S. Xing, F. Liu, Q. Wang, X. Zhao, and T. Li, "A hierarchical attention model for rating prediction by leveraging user and product reviews," *Neurocomputing*, vol. 332, pp. 417–427, 2019.
- [15] C. Courcoubetis and R. Weber, "Pricing communication networks: Economics," in *Technology and Modelling*. Hoboken, NJ, USA: Wiley, 2003.
- [16] A. N. Quttoum, H. Otkrok, and Z. Dziong, "A collusion-resistant mechanism for autonomic resource management in virtual private networks," *Comput. Commun.*, vol. 33, no. 17, pp. 2070–2078, Nov. 2010.
- [17] Y. M. Chen, "Research on SDN-based multi-tenant data center network virtualization method," M.S. thesis, Beijing Univ. Posts Telecommun., Beijing, China, 2016.
- [18] P. Y. Zhang, F. Liu, G. Aujla, and S. Vashisht, "VNE strategy based on chaos hybrid flower pollination algorithm considering multi-criteria decision making," *Neural Comput. Appl.*, to be published.
- [19] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci.*, vol. 450, pp. 200–226, Jun. 2018.
- [20] P. Y. 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.-Practice Exper.*, 2020.
- [21] X. Zheng, J. Tian, X. Xiao, X. Cui, and X. Yu, "A heuristic survivable virtual network mapping algorithm," *Soft Comput.*, vol. 23, no. 5, pp. 1453–1463, Mar. 2019.
- [22] X. Xiao, X. Zheng, and Y. Zhang, "A multidomain survivable virtual network mapping algorithm," *Secur. Commun. Netw.*, vol. 2017, pp. 1–12, 2017.
- [23] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proc. 1st ACM Workshop Virtualized Infrastruct. Syst. Archit. ACM*, pp. 81–88, 2009.
- [24] H. L. Li, L. Zou, and X. Wei, "Virtual network embedding algorithm based on improved sub-graph isomorphism search," *J. Jilin Univ.*, vol. 43, no. 1, pp. 165–171, 2013.
- [25] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38–47, Apr. 2011.
- [26] J. Tian, H. Zhang, D. Wu, and D. Yuan, "QoS-constrained medium access probability optimization in wireless interference-limited networks," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1064–1077, Mar. 2018.
- [27] X. W. Zheng, B. Hu, D. Lu, Z. Chen, and H. Liu, "Energy-efficient virtual network embedding in networks for cloud computing," *Int. J. Web Grid Services*, vol. 13, no. 1, pp. 75–93, 2017.
- [28] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. IEEE INFOCOM 28th Conf. Comput. Commun.*, Apr. 2009, pp. 783–791.
- [29] C. Ji, C. Zhao, S. Liu, C. Yang, L. Pan, L. Wu, and X. Meng, "A fast shapelet selection algorithm for time series classification," *Comput. Netw.*, vol. 148, pp. 231–240, 2019.
- [30] W. Rankothge, F. Le, A. Russo, and J. Lobo, "Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 343–356, Jun. 2017.
- [31] T. Wang and M. Hamdi, "Presto: Towards efficient online virtual network embedding in virtualized cloud data centers," *Comput. Netw.*, vol. 106, pp. 196–208, Sep. 2016.
- [32] M. He, L. Zhuang, S. Tian, and G. K. W. Zhang, "DROI: Energy-efficient virtual network embedding algorithm based on dynamic regions of interest," *Comput. Netw.*, vol. 166, pp. 1–11, Jan. 2020.
- [33] S. Wang, J. Bi, J. Wu, A. V. Vasilakos, and Q. Fan, "VNE-TD: A virtual network embedding algorithm based on temporal-difference learning," *Comput. Netw.*, vol. 161, pp. 251–263, Oct. 2019.
- [34] C. Zhao and B. Parhami, "Virtual network embedding through graph eigenspace alignment," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 632–646, Jun. 2019.
- [35] Z. Yan, J. Ge, Y. Wu, H. Zheng, L. Li, and T. Li, "Automatic virtual network embedding based on deep reinforcement learning," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun. IEEE 17th Int. Conf. Smart City; IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 625–631.
- [36] Y. Yuan, Z. Tian, C. Wang, F. Zheng, and Y. Lv, "A Q-learning-based approach for virtual network embedding in data center," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 1995–2004, Apr. 2020.
- [37] C. Li, S. Yang, T. T. Nguyen, E. L. Yu, and X. Yao, "Benchmark generator for CEC 2009 competition on dynamic optimization," *Sens. Lett.*, vol. 11, pp. 900–909, Oct. 2008.
- [38] R. Mijumbi, J.-L. Gorricho, J. Serrat, M. Shen, K. Xu, and K. Yang, "A neuro-fuzzy approach to self-management of virtual network resources," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1376–1390, Feb. 2015.
- [39] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.
- [40] 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.
- [41] S. Fickas and M. S. Feather, "Requirements monitoring in dynamic environments," in *Proc. IEEE Int. Symp. Requirements Eng. (RE95)*, Mar. 1995, pp. 140–147.
- [42] A. Oveis and N. Amjadi, "Net requirement prediction for power systems by a new neural network-based forecasting engine," *Complex*, vol. 21, no. 2, pp. 296–308, 2016.
- [43] L. V. Ballestra and G. Pacelli, "Pricing European and American options with two stochastic factors: A highly efficient radial basis function approach," *J. Econ. Dyn. Control*, vol. 37, no. 6, pp. 1142–1167, Jun. 2013.
- [44] X. Ma, Y. Jin, and Q. Dong, "A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting," *Appl. Soft Comput.*, vol. 54, pp. 296–312, May 2017.
- [45] E. Costa, S. Porziani, C. Groth, M. E. Biancolini, E. Costa, and K. C. Giannakoglou, "Aerodynamic optimization of car shapes using the continuous adjoint method and an RBF Morph," *Adv. Evol. Deterministic Methods Design, Optim. Control Eng. Sci.*, pp. 173–187, Jul. 2015.
- [46] *The 45th Statistical Report on Internet Development in China in 2020—Internet Access Environment*. [Online]. Available: <http://www.199it.com/archives/1042095.html>
- [47] X. Yu, H. Wang, X. Zheng, and Y. Wang, "Effective algorithms for vertical mining probabilistic frequent patterns in uncertain mobile environments," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 23, nos. 3–4, pp. 137–151, 2016.
- [48] X. W. Zheng, Y. Yu, Y. Li, and H. Liu, "An enhanced multi-objective group search optimizer based on multi-producer and crossover operator," *J. Inf. Sci. Eng.*, vol. 33, no. 1, pp. 37–50, 2017.

- [49] X. W. Zheng, B. Hu, D. Lu, and H. Liu, "A multi-objective virtual network embedding algorithm in cloud computing," *J. Internet Technol.*, vol. 4, no. 17, pp. 687–701, 2016.



XIAN-CUI XIAO received the M.E. degree from the School of Information Science and Engineering, Shandong Normal University, China, in 2017, where she is currently pursuing the Ph.D. degree. Her research interests include intelligent algorithm, computer networks, and cloud computing.



XIANG-WEI ZHENG received the B.S. and M.E. degrees in computer science and the Ph.D. degree in management science and engineering from Shandong Normal University, in 1995, 1998, and 2008, respectively. He is currently a Professor with the School of Information Science and Engineering, Shandong Normal University, China. He has been involved in several national natural science foundation of China. He is the author of more than 50 national and international publications in conferences and journals. His research interests include cloud computing, computer networks, and computational intelligence.



YI WEI received the B.S. degree in software engineering and the Ph.D. degree in computer science and technology from Shandong University, in 2013 and 2019, respectively. She is currently a Lecturer with the School of Information Science and Engineering, Shandong Normal University, China. Her research interests include cloud computing and artificial intelligence.



XIN-CHUN CUI received the B.E. degree from Shandong Normal University, in 1995, and the M.E. degree from Shanghai Normal University, in 2002, and the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics. He is currently a Professor with the School of Information Science and Engineering, Qufu Normal University, China. His research interests include information security and computational neuroimaging.

• • •