

Research Article

Multiobjective Model and Improved Artificial Raindrop Algorithm for Virtual Network Mapping

Hejun Xuan ¹, Xuelin Zhao,¹ Lei You ^{1,2}, Zhenghui Liu,^{1,2} and Yanling Li^{1,2}

¹School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China

²Henan Key Lab. of Analysis and Application of Education Big Data, Xinyang Normal University, Xinyang 464000, China

Correspondence should be addressed to Lei You; lei_perfect_xynu@126.com

Received 9 February 2021; Revised 21 February 2021; Accepted 26 February 2021; Published 12 March 2021

Academic Editor: Chi-Hua Chen

Copyright © 2021 Hejun Xuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network virtualization (NV) can provide the resource according to the request and can improve the flexibility of the network. It has become the key technology of 5G communication. Resource scheduling in virtual network mapping is an important problem faced by NV technology. To enhance the performance of the network, optimal resource scheduling schemes should be determined. In general, maximum index of frequency slots used, energy consumption, and the ratio of used frequency slots are three very important indicators for the network. Based on the previous research studies, we first take these three indicators into the objective of the mathematical model and define a novel multiobjective optimization model. Then, the three objectives are integrated into one objective to be minimized by using the sum weighted strategy. Finally, an efficient algorithm, which integrates the advantages of artificial raindrop algorithm (ARA), particle swarm optimization, and differential evolution, is proposed and denoted as IARA. Moreover, simulation experiments have been conducted in several experimental scenes with other compared algorithms. The experimental results show that the proposed algorithm IARA can obtain better scheduling schemes than the compared algorithms.

1. Introduction

Network virtualization is actually to realize that multiple heterogeneous virtual networks can run in a basic network. By creating a realistic and stable virtual environment, resources reuse of routers, links, and other physical devices can be realized, which greatly improves resource utilization, thus providing support for Internet of Things, cloud computing, and other technologies [1, 2]. Cloud platform is based on the underlying infrastructure of network virtualization technology to achieve the resource sharing, and distributed architecture and the characteristic of being extensible also make cloud platform able to improve the simulation of network scale, but the establishment of the virtual network and dismantling make the demand of the network business change, in order to make the underlying network support heterogeneous virtual network request as much as possible and achieve the efficient scheduling and management of resources [3]. What is more, network virtualization can

accelerate innovation of network architecture. Now there are many different kinds of resources in the network environment, different function, virtual resource distribution is not the same, network heterogeneity and dynamics. The virtual nodes deployed in location and resources to meet the basic requirements of physical nodes and to ensure meeting the resource constraint conditions such as virtual nodes in the path of the physical nodes are linked together, the core problem is the virtual network mapping, there is also an urgent need to solve the key challenge, and the field has become a very popular area of research [4, 5]; it enables network operators to share a physical network in case of different operating virtual optical network and the optical layer to simplify resource management, providing flexible spectrum allocation schemes and secure application services [6]. [7]. Network virtualization as solving the problem of network “rigidity” provides an effective means; it can realize multiple heterogeneous virtual networks sharing the underlying network resources; it not only improves the

utilization rate of network resources but also brings more flexible services; virtual network mapping, as one of the key technologies of network virtualization, has been widely concerned [8–10].

Virtual network mapping has been proved to be an NP-hard problem, and the related solutions have become a research hotspot [11, 12]. Virtual network mapping can be divided into one stage and two stages according to the steps to map, a phase mapping algorithm because of high complexity, low adaptability, now two phase mapping algorithm is the mainstream of the study is the first network all nodes mapping, mapping, and then to link so the node mapping strategy for virtual network mapping plays a key role in success [13]. Node mapping methods mainly include heuristic algorithm [14], linear programming solution [15], and intelligent optimization algorithm [16]. Node sorting has also developed from simple node resource sorting [17] to topological resource joint sorting [18], gradually forming the problem of node multi-index sorting [19]. The authors of [20] put forward a kind of node importance evaluation method based on TOPSIS, and, on the basis of considering the node resources property, the introduction of social network analysis method of the centrality, and approaching degrees centrality, Euclidean distance of TOPSIS method cannot distinguish with positive and negative ideal solution distance equal points, index weight, and node set artificially; subjectivity is stronger. The study in [21] proposed a node ranking method with weighted relative entropy, which can realize the joint perception of virtual topology and physical topology, but its index weight needs to be adjusted manually according to environmental changes, so it relies heavily on experience. The study in [22] introduced the extraction process of the underlying network features and the node sorting method and used the particle swarm optimization method to train the index weight vector. It can automatically determine the parameters in the algorithm for different optimization objectives, but the algorithm complexity is high. The study in [23] proposed a virtual network mapping algorithm based on greedy algorithm. In the node mapping stage, only the CPU resources of the nodes and the bandwidth of the adjacent links were considered, without considering the topological attributes. In order to further improve the mapping efficiency, scholars improve the performance of node mapping algorithm. The study in [24] proposed the optimization model of energy consumption and applied it to the problem of virtual network mapping. The study in [25] designs an optimal backup topology for single link failure in virtual SDN, which improves resource utilization and shortens algorithm execution time. The study in [26] focuses on the problem of network forwarding resources and load balancing and designs an integer linear programming model to solve the problem of online resource allocation of multiple virtual links in SDN environment. The study in [27] proposes a time-sensitive virtualization controller deployment algorithm for SDN controller network delay, which reduces the communication delay between the switch and the controller. However, there are few researches on combinatorial optimization problems, which aim to control delay and balance network load.

In general, maximum index of used frequency slots, energy consumption, and the ratio of used frequency slots are three very important indicators for the network. Based on the previous researches, we first take these three indicators into the objective of the mathematical model and define a novel multiobjective optimization model. Then, the three objectives are integrated into one objective to be minimized by using the sum weighted strategy. Finally, an efficient algorithm, which integrates the advantages of artificial raindrop algorithm (ARA), particle swarm optimization, and differential evolution, is proposed and denoted as IARA. We summarize the major contributions of this paper as follows:

- (i) We first take the three indicators into the objective of the mathematical model and define a novel multi-objective optimization model
- (ii) An efficient algorithm, which integrates the advantages of artificial raindrop algorithm (ARA), particle swarm optimization, and differential evolution, is proposed

The rest of the paper is organized as follows: Section 2 describes the problem and establishes a novel three-objective optimization model. To solve the established mathematical model with high performance, we give the overview of the artificial raindrop algorithm (ARA) and propose an improved artificial raindrop algorithm (IARA) in Section 3. Section 4 presents experimental results and analysis. Conclusions with a summary are drawn in Section 5.

2. Problem Formulation

2.1. Problem Description. Directed graph $G = (V, E)$ denotes a physical network topology. $V = \{v_i | i = 1, 2, \dots, N\}$ is the network nodes set, where N and v_i ($i = 1, 2, \dots, N$) are the numbers of nodes and the optical nodes, respectively. For the node v_i in physical network, there are $c(v_i)$ virtual machines (VMs) in it. $E = \{l_{ij} | i, j \in V\}$ represents optical fiber links set, and $|E|$ is the link number of a network. l_{ij} ($1 \leq i \neq j \leq N$) is the link between physical nodes v_i and v_j in the network topology. Let $F = \{f_u | u = 1, 2, \dots, |F|\}$ denote the set of available frequency slots in each link, and let $|F|$ be the number of frequency slots.

$VON = \{VON^1, VON^2, \dots, VON^M\}$ denotes the set of virtual optical networks (VONs) in network G , where M represents the number of VONs. $V^m = \{v_1^m, v_2^m, \dots, v_{N_m}^m\}$ denotes the set of virtual nodes (VNs) in VON^m , where N_m is the number of VN i in VON^m . In general, we have $N_m \leq N$. Ω_n^m denotes the candidate set v_n^m ($n = 1, 2, \dots, N_m$; $m = 1, 2, \dots, M$) can map to, and the nodes in Ω_n^m are all adjacent to each other. $R^m = \{r_1^m, r_2^m, \dots, r_{|R^m|}^m\}$ is the set of virtual connection requests (VCR) in VON^m . r_k^m ($k = 1, 2, \dots, |R^m|$; $m = 1, 2, \dots, M$) denotes the k^{th} VCR in VON^m . For r_k^m in VON^m , it can be described as $r_k^m = (s_k^m, d_k^m, T_k^m)$. Source node, destination node, and required capacity are s_k^m , d_k^m , and T_k^m , respectively.

For a specific virtual connection request $r_k^m = (s_k^m, d_k^m, T_k^m)$, if VN s_k^m and d_k^m are mapping to nodes $s_{k'}$ and $d_{k'}$ ($s_{k'}, d_{k'} \in V$), respectively, r_k^m would translate into a

physical connection request $r_{k'} = (s_{k'}, d_{k'}, T_{k'})$, where $T_{k'} = T_k^m$. When all the VNs were mapped to nodes, all the VONs are mapping to the network. That is to say, all the VCR are translated to the physical connection requests. $R' = \{r_1, r_2, \dots, r_{|R'|}\}$ denotes the physical connection requests set. Thus, $|R'| = \sum_{m=1}^M |R^m|$ denotes the number of VCRs.

2.2. Multiobjective Mathematical Modeling

2.2.1. Objective Functions

- (1) The first objective is minimizing the maximum index of frequency slots used. The maximum index of frequency slots used is calculated by

$$F_{\max} = \max_{l_{ij} \in E} \left\{ |F_{l_{ij}}| \right\}, \quad (1)$$

where $|F_{l_{ij}}|$ is the maximum index of frequency slots used on l_{ij} . Since $F_{\max} \leq |F|$, the maximum index of frequency slots used can be normalized as

$$f_1 = \frac{\max_{l_{ij} \in E} \left\{ |F_{l_{ij}}| \right\}}{|F|}. \quad (2)$$

Thus, we have $0 \leq f_1 \leq 1$, and the first objective function can be represented as

$$\min f_1 = \min \left\{ \frac{\max_{l_{ij} \in E} \left\{ |F_{l_{ij}}| \right\}}{|F|} \right\}. \quad (3)$$

- (2) The energy consumption is minimized in second objective. Energy consumption for all the connection requests can be calculated by

$$E_{\text{total}} = \sum_{k'=1}^{|R'|} (\lambda_{k'}^q g(Q_{k'}^q)), \quad (4)$$

where $g(Q_{k'}^q)$ denotes the energy consumption of $r_{k'}$ when it occupies the path $Q_{k'}^q$. $\lambda_{k'}^q$ is a Boolean variable; if $r_{k'}$ occupies the q^{th} path in $Q_{k'}$, $\lambda_{k'}^q = 1$; otherwise, $\lambda_{k'}^q = 0$.

Since $\sum_{k'=1}^{|R'|} (\lambda_{k'}^q g(Q_{k'}^q)) \leq \sum_{k'=1}^{|R'|} (g(Q_{k'}^q))$, the total energy consumption can be normalized as

$$f_2 = \frac{\sum_{k'=1}^{|R'|} (\lambda_{k'}^q g(Q_{k'}^q))}{\sum_{k'=1}^{|R'|} (g(Q_{k'}^q))}. \quad (5)$$

Similar to the first objective, we have $0 \leq f_2 \leq 1$. Thus, objective function can be expressed by

$$\min f_2 = \min \left\{ \frac{\sum_{k'=1}^{|R'|} (\lambda_{k'}^q g(Q_{k'}^q))}{\sum_{k'=1}^{|R'|} (g(Q_{k'}^q))} \right\}. \quad (6)$$

- (3) The third objective is maximizing the ratio of frequency slots utilization, and it is defined as

$$\text{RFSU} = \frac{\text{FS}_{\text{total}}}{\text{NL}_{\text{used}} \times |F|}, \quad (7)$$

where FS_{total} and NL_{used} are the total FSs used and number of links used, respectively. Since $\text{FS}_{\text{total}} \leq \text{NL}_{\text{used}} \times |F|$, $0 \leq \text{RFSU} \leq 1$. So, this objective can be expressed by

$$\max \text{RFSU} = \max \left\{ \frac{\text{FS}_{\text{total}}}{\text{NL}_{\text{used}} \times |F|} \right\}. \quad (8)$$

Since the first and second objectives are minimized, we can rewrite the third objective as

$$\min f_3 = 1 - \max \left\{ \frac{\text{FS}_{\text{total}}}{\text{NL}_{\text{used}} \times |F|} \right\}. \quad (9)$$

Obviously, $0 \leq f_3 \leq 1$.

The three objectives can be integrated into one to be minimized by sum weighted strategy; that is,

$$\min f = \min \{ \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 \}, \quad (10)$$

where α_1 , α_2 , and α_3 are three weights to adjust the importance of the three objectives, and we have $0 \leq \alpha_1, \alpha_2, \alpha_3 \leq 1$ and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Since $0 \leq f_1, f_2, f_3 \leq 1$, $0 \leq f \leq 1$. In addition, some constraint conditions should be satisfied. These constraint conditions are given in our previous paper [28]. To solve the multiobjective model, an efficient algorithm, which integrates the advantages of artificial raindrop algorithm (ARA), particle swarm optimization, and differential evolution, is proposed.

3. Overview of Artificial Raindrop Algorithm and Improved Artificial Raindrop Algorithm

3.1. Overview of Artificial Raindrop Algorithm. Artificial raindrop algorithm (ARA) is a kind of nonbiological heuristic algorithm based on population. The algorithm is inspired by the phenomenon of natural rainfall and transplants the effective information processing mechanism contained in the process of natural rainfall into the optimization algorithm design. According to the observation of the natural rainfall process, the whole optimization design cycle is divided into six stages: raindrop formation, raindrop descent, raindrop collision, raindrop flow, raindrop pool renewal, and water vapor renewal. During the optimization process, the location of water vapor or raindrops generated was evaluated using altitude, and the lower elevation positions were recorded in the raindrop pool.

Similar to the most metaheuristic algorithms, ARA searches the optimal solution with an initial population by randomly generating N_p vapors in a limited search space, and each vapor has a corresponding position defined as $v_i = (v_i^1, v_i^d, \dots, v_i^d, \dots, v_i^D)$ $i = 1, 2, \dots, N_p$, where N_p is the population size, D is the dimension of problem, and v_i^d is the position of the i -th vapor in the d -th dimension.

3.1.1. Raindrop Generation. Generally speaking, raindrops are produced by constantly absorbing surrounding water vapor. For simplicity, assume the position of raindrops water

vapor around the geometric center. Therefore, its location can be defined as $RP = (x^1, x^2, \dots, x^d, \dots, x^D)$, where $x^d = (1/N_p) \sum_{i=1}^{N_p} x_i^d$.

3.1.2. Raindrop Descent. When ignoring the impact of external factors, free-falling raindrops from the clouds reach the ground. This means that a component has changed the location of raindrops, and the raindrops will move to a new location representing new raindrops. For raindrops RP, x^{d_j} raindrops at d_j -position on the first dimension, wherein d_j ($j = 1, 2, 3, 4$) from $\{1, 2, \dots, D\}$ set arbitrarily selected. Thus, new raindrop (NRP = $(y^1, y^2, \dots, y^d, \dots, y^D)$), d -th dimension of y^d can be from x^{d_2}, x^{d_3} , and x^{d_4} linear combinations obtained, defined as follows:

$$y^d = \begin{cases} x^{d_2} + \gamma(x^{d_3} - x^{d_4}), & d = d_1, \\ x^d, & \text{otherwise,} \end{cases} \quad (11)$$

where γ is a random number in $(-1, 1)$ and $d = 1, 2, \dots, D$.

3.1.3. Raindrop Collision. When a new raindrop hits the ground, it will split into a large number of small raindrops. These small raindrops (SRP_i , ($i = 1, 2, \dots, N_s$)) fly in all directions. Accordingly, SRP_i can be defined as

$$SRP_i = NRP + \text{sign}(\alpha - 0.5) \cdot \lg(\beta) \cdot (NRP - v_i), \quad (12)$$

where k , which is randomly selected from the set $1, 2, \dots, N_p$, is an index. α and β are two random numbers and are distributed in the range $(0, 1)$ uniformly. $\text{sign}(\cdot)$ represents the sign function.

3.1.4. Raindrop Flowing. Under gravity, SRP_i ($i = 1, 2, \dots, N_p$) will flow to the low altitude and will most eventually stop at a lower altitude from a height position (i.e., a better solution). In the process of evolutionary algorithms, these better solutions can provide more information about the direction of promising progress. Therefore, the design comprising N_{RPO} raindrops pool (textbf RPO) to track these lower positions is found during the search, and textbf RPO update: (1) textbf RPO any feasible solution is to initialize the search space; (2) after each iteration, the optimal solution is added in the current population textbf RPO; and (3) if the size exceeds a threshold RPO value set in advance, in order to maintain the textbf RPO of size stability, reduce the amount of calculation, some solutions of \textbf{RPO} are deleted randomly

What is more, the flowing direction of raindrop $dSRP_i$ for SRP_i ($i = 1, 2, \dots, N_p$) can be constructed based on the linear combination of two vectors $dSRP1_i$ and $dSRP2_i$ as follows:

$$dSRP_i = \tau_1 \cdot \theta_i^1 \cdot dSRP1_i + \tau_2 \cdot \theta_i^2 \cdot dSRP2_i, \quad (13)$$

where τ_1 and τ_2 are two step parameters of SRP_i flowing, and θ_i^1 and θ_i^2 are generated randomly in the range $(0, 1)$. $dSRP1_i$ and $dSRP2_i$ are two vectors and they are defined as follows:

$$\begin{aligned} dSRP1_i &= \text{sign}(F(RPO_{k_1}) - F(SRP_i)) \cdot (RPO_{k_1} - SRP_i), \\ dSRP2_i &= \text{sign}(F(RPO_{k_2}) - F(SRP_i)) \cdot (RPO_{k_2} - SRP_i), \end{aligned} \quad (14)$$

where RPO_{k_1} and RPO_{k_2} ($1 \leq k_1 \neq k_2 \leq N_{rpo}$) are any two of candidate solutions in **RPO**. $F(\cdot)$ denotes the fitness function. Thus, the i -th new small raindrop ($NSRP_i$) can be defined as

$$NSRP_i = SRP_i + dSRP_i. \quad (15)$$

In general, it is necessary to introduce a parameter N_{MF} to control the maximum flow rate. Thus, these new small raindrops will stay in the locations with a lower elevation or evaporate after several flowing because of the parameter N_{MF} .

3.1.5. Vapor Updating. In order to converge ARA, in the water vapor update process, select a new ranking method using the small raindrop and a small steam of N_p as the next best solution vapor population.

3.2. Improved Artificial Raindrop Algorithm (IARA)

3.2.1. Encoding. Four steps are used to solve the problems of mapping VON: (1) physically mapping a virtual node to node, (2) connection request sorting, (3) routing, and (4) spectrum allocation. We randomly sort connection which does not require coding. Accordingly, it only needs to be coded in steps 1, 3, and 4. Therefore, using the virtual node mapping populations, populations and spectrum allocation routing population of these three individuals into the population of these three steps are necessary and reasonable. This paper presents the virtual node mapping and routing of coding schemes and population initialization program [28].

Each gene represents the starting frequency index for each connection request slot on the chromosome. Routing similar population of chromosomes, the chromosome length is carded quantity of connection requests. For example, $z = (z_{k'})_{1 \text{ multiplied } N_{R_i}}$ is a spectral distribution in the population of chromosomes. If $z_{k'} = u$, then u -th is allocated to the $(u + B_{k'} + GF - 1)$ -th frequency slot of the connection request $r_{k'}$. The population of spectrum allocation has been initialized in Algorithm 1.

3.2.2. Improved Raindrop Flowing. In the improved algorithm, flowing direction of raindrop $dSRP_i$ of SRP_i ($i = 1, 2, \dots, N_p$) can be constructed based on the linear combination of three vectors $dSRP1_i$, $dSRP2_i$, and $dSRP3_i$ as follows:

$$dSRP_i = \tau_1 \cdot \theta_i^1 \cdot dSRP1_i + \tau_2 \cdot \theta_i^2 \cdot dSRP2_i + \tau_3 \cdot \theta_i^3 \cdot dSRP3_i, \quad (16)$$

where τ_3 is a step parameter of SRP_i like τ_1 and τ_2 and θ_i^3 also is generated randomly in the range $(0, 1)$. $dSRP3_i = (dSRP3_i^1, dSRP3_i^2, \dots, dSRP3_i^d, \dots, dSRP3_i^D)$ is a vector and it is defined as follows:

<p>Input: $Q_{k'}$ for all physical connection requests R', Population size Pop_{Size}</p> <p>Output: Spectrum assignment population RP</p> <ol style="list-style-type: none"> (1) for $p = 1$ to Pop_{Size} do <li style="padding-left: 20px;">(2) for $k' = 1$ to R' do <li style="padding-left: 40px;">(3) An integer u is generated randomly between 1 and F; <li style="padding-left: 40px;">(4) $\text{SP}(p, k') = u$; <li style="padding-left: 20px;">(5) end (6) end

ALGORITHM 1: Spectrum assignment population initialization.

$$\begin{aligned} \text{dSRP}_i^d = & \frac{\text{SRP}_{i,\beta}^d + \text{SRP}_{i,\beta}^d + \text{SRP}_{i,\beta}^d}{3} \\ & + r_3 \cdot (\text{SRP}_{i,\text{best}}^d - \text{SRP}_i^d) + r_4 \cdot (\text{SRP}_j^d - \text{SRP}_i^d), \end{aligned} \quad (17)$$

where r_3, r_4 , and r_5 are three random vectors in $[0, 1]$ and $\text{SRP}_{i,\text{best}}^d$ denotes the best position of i -th raindrop in the past. $\text{SRP}_{i,\beta}^d$, $\text{SRP}_{i,\gamma}^d$, and $\text{SRP}_{i,\delta}^d$ are defined as

$$\begin{aligned} \text{SRP}_{i,\beta}^d &= \text{SRP}_{\beta}^d - a_1 \cdot |a_2 \cdot \text{SRP}_{\beta}^d - \text{SRP}_i^d|, \\ \text{SRP}_{i,\gamma}^d &= \text{SRP}_{\gamma}^d - a_1 \cdot |a_2 \cdot \text{SRP}_{\gamma}^d - \text{SRP}_i^d|, \\ \text{SRP}_{i,\delta}^d &= \text{SRP}_{\delta}^d - a_1 \cdot |a_2 \cdot \text{SRP}_{\delta}^d - \text{SRP}_i^d|, \end{aligned} \quad (18)$$

where a_1 and a_2 are two random numbers in the range $(0, 1)$; SRP_{β}^d , SRP_{γ}^d , and SRP_{δ}^d denote the optimal raindrop, sub-optimum raindrop, and third-optimum raindrop.

4. Experiments and Analysis

In order to prove the effectiveness and efficiency of the proposed algorithm, this section performs several experiments and results are given. In Section 4.1, the parameters used in the algorithm are given. The results are given in Section 4.2. Finally, Section 4.3 presents the analysis of experimental results.

4.1. Parameters Setting. In our simulation experiments, there are three network topologies. NSFNET topology includes 14 nodes and 21 links. CHNNET topology includes 15 nodes and 27 links. ARPANET topology includes 20 nodes and 32 links. Subcarrier modulation level uses 4 kinds: BPSK, QPSK, 8QAM, and 16QAM. Thus, for BPSK, QPSK, 8QAM, and 16QAM different modulation levels, ML 2, 3, and 4 may be selected. The transmission distance of modulation levels are 9600, 4800, 2400, and 1200 km [29]. We assume that each frequency slot is 2.5 GHz. Furthermore, the network topology has 5–10 VMs on each physical node. Each has 4 or 6 VON virtual nodes, a virtual connection request is present between a pair of virtual nodes, and the probability is 0.5. Each virtual node in VONs has between 2–4 candidate nodes in EONs.

4.2. Experimental Results. In order to verify the performance of the proposed algorithm, we compared the proposed algorithm IARA with the ARA proposed in the literature. In addition, we compared the IARA scheme with the DC & TP-EA scheme and considered the energy saving of DCs in node mapping and the energy saving of TPs in link mapping (proposed in [30]) to verify the performance of the IARA scheme. In addition, we also compared the proposed algorithm with the three best-performing algorithms for solving the VONs mapping problem. The first algorithm is expressed as CAN-A and is proposed in [31]. In order to make it more suitable for real networks, CAN-A considers four types of node and link constraints and constructs a subset of candidate substrate nodes and a subset of candidate substrate paths before embedding. LSCD algorithm is map the largest bandwidth requirement virtual links on the shortest distance physical links. The last comparison algorithm is GRC-SVNE proposed in the literature. In the node mapping phase, GRC-SVNE selects some nodes as candidate nodes according to the mapping capacity of all nodes. Then Dijkstra's algorithm is used for link mapping in the second stage.

In order to verify the performance of the algorithm, there are two different experiments to select scenes. In the first experiment scenario, each virtual optical network's virtual section was fixed to 5 points. There is a request to a virtual connection between each pair of virtual nodes, and the probability is 0.5. The number of virtual optical networks from 10 to 20 is selected. In Figures 1–4, the results are given objective function with three network VONs quantity change algorithms. When $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$, the experimental results obtained by the algorithm are shown in Figure 1. $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0$. The results obtained by the algorithm are shown in Figure 2. Figure 3 shows $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$. Figure 4 shows $\alpha_1 = 1/3, \alpha_2 = 1/3, \alpha_3 = 1/3$.

In a second experiment scenario, the number of fixed VONs $M = 30$. In each virtual optical network, the number of virtual nodes is from 3 to 7 (Figures 5–8). The results are given objective function with virtual three network nodes change algorithm. $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$. The results obtained by the algorithm are shown in Figure 5. When $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0$, the experimental results obtained by the algorithm are shown in Figure 6. Figure 7 shows experimental results obtained by the algorithm when

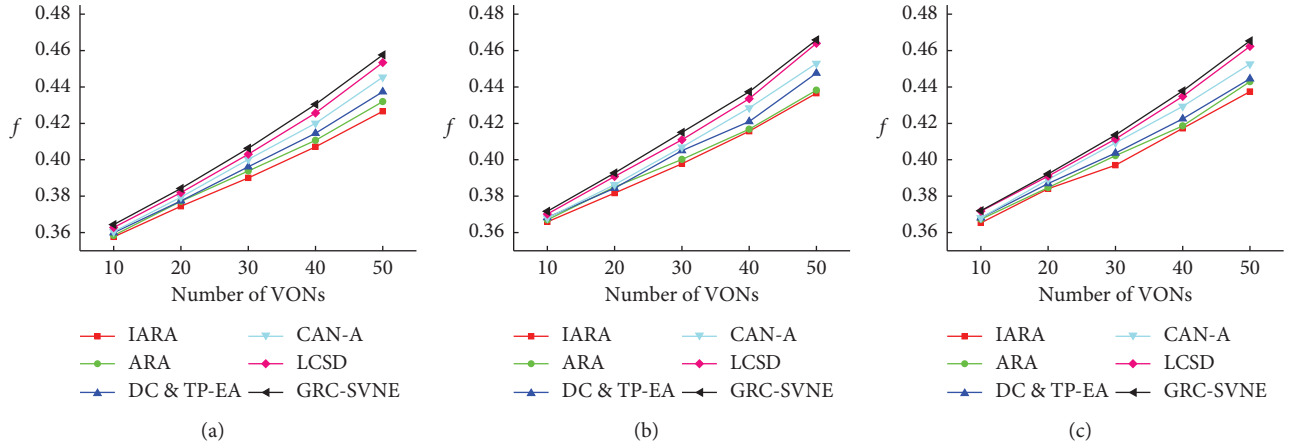


FIGURE 1: Experimental results in the first scene when $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

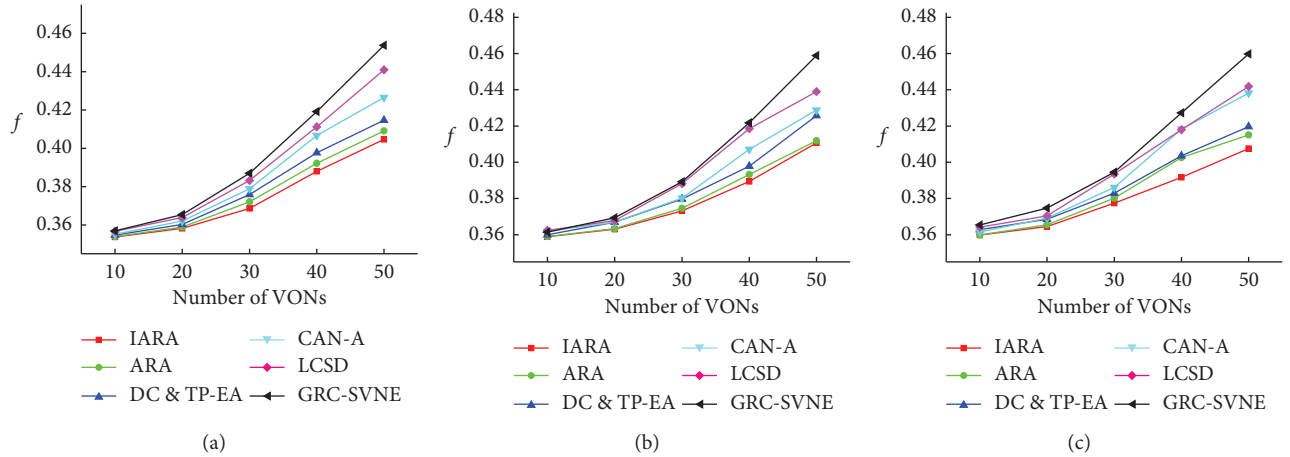


FIGURE 2: Experimental results in the first scene when $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 1$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

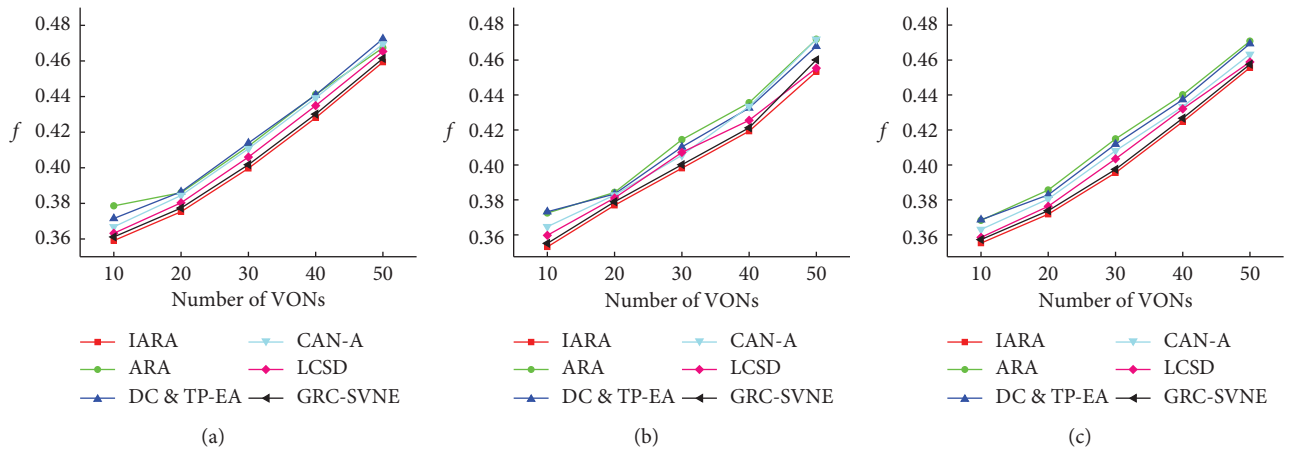


FIGURE 3: Experimental results in the first scene when $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

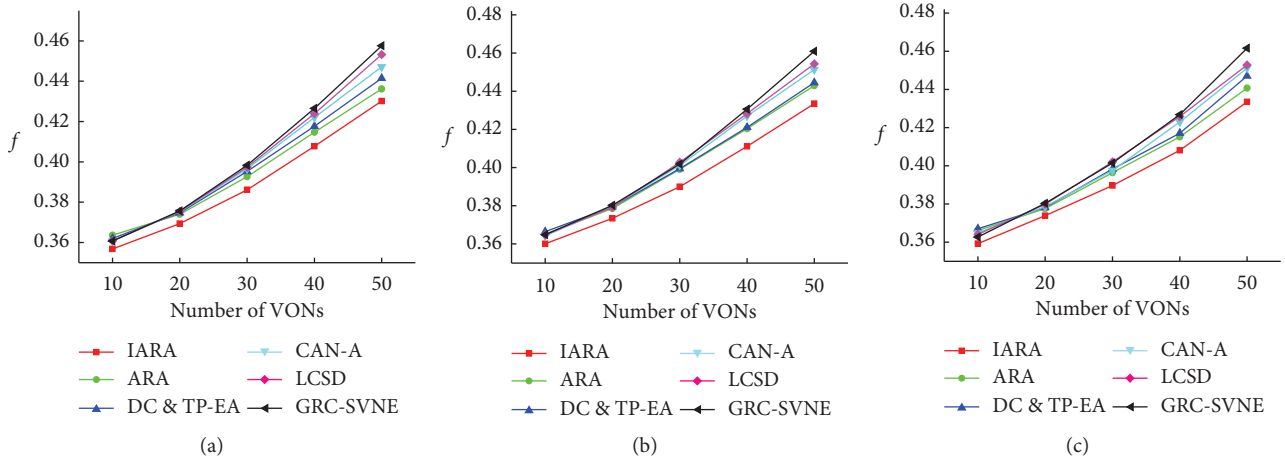


FIGURE 4: Experimental results in the first scene when $\alpha_1 = 1/3, \alpha_2 = 1/3, \alpha_3 = 1/3$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

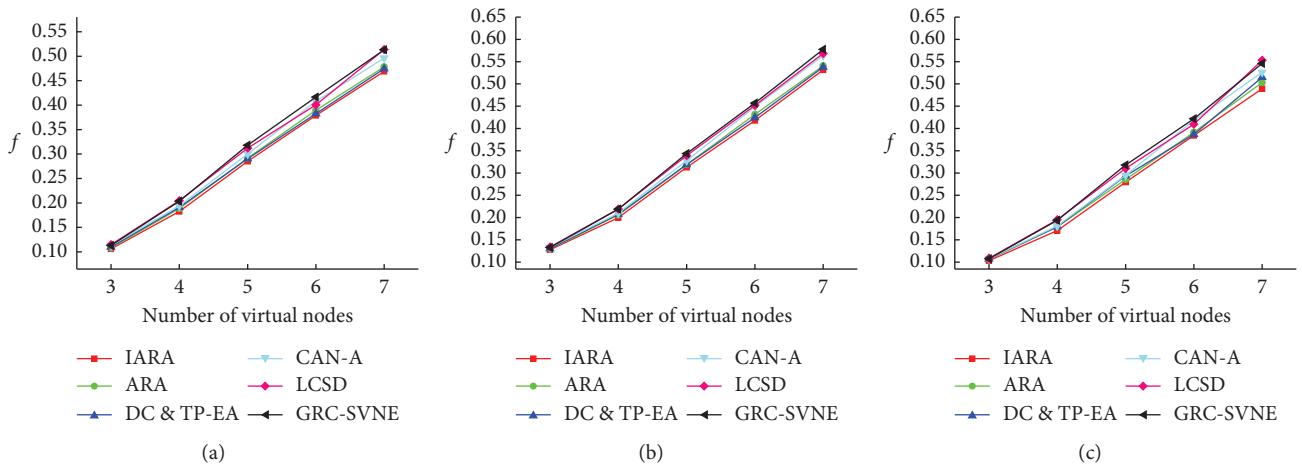


FIGURE 5: Experimental results in the second scene when $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

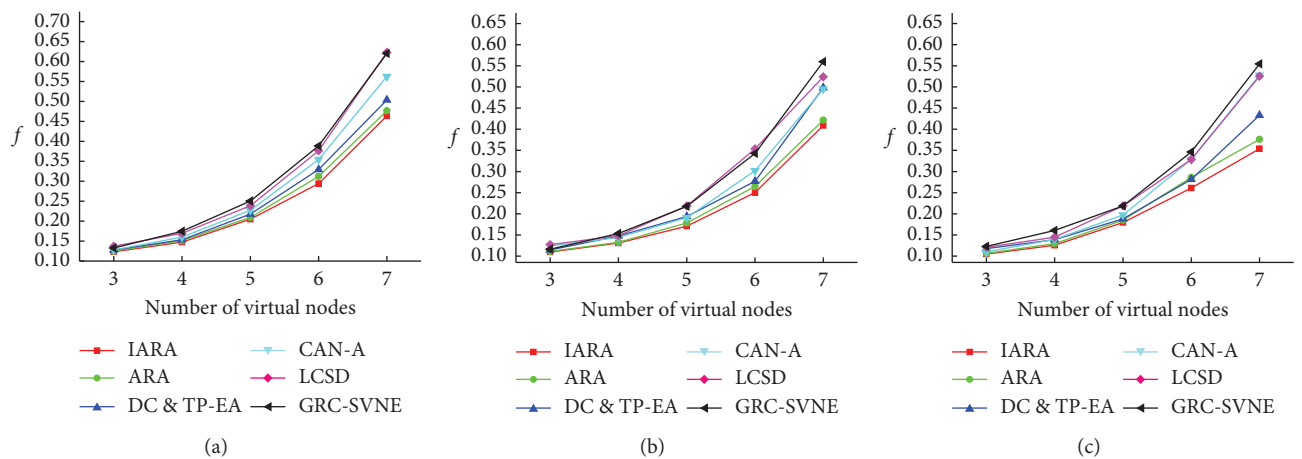


FIGURE 6: Experimental results in the second scene when $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 1$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

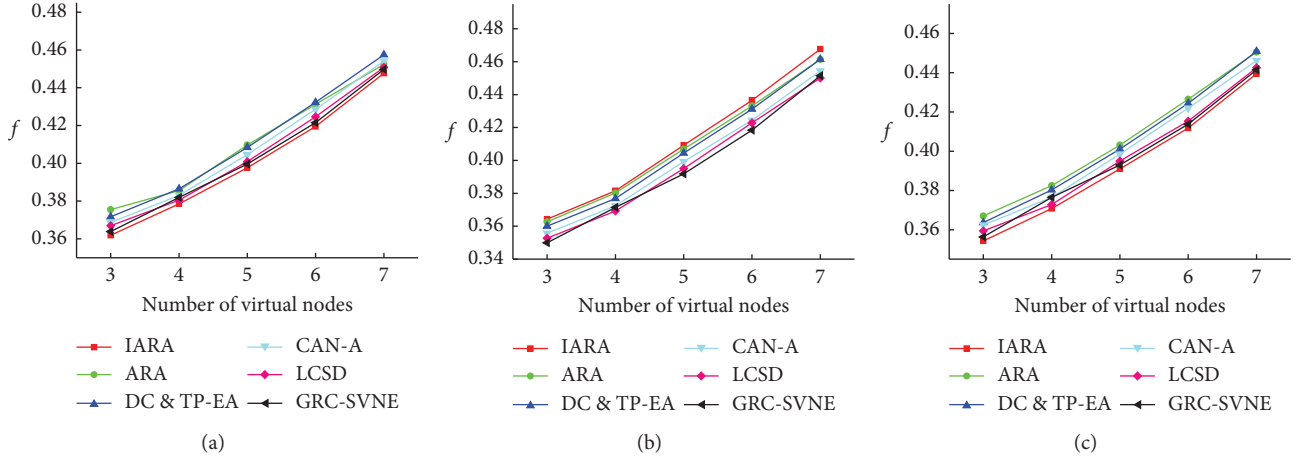


FIGURE 7: Experimental results in the second scene when $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

$\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$. Figure 8 shows the experimental results obtained when $\alpha_1 = 1/3, \alpha_2 = 1/3, \alpha_3 = 1/3$.

4.3. Experimental Analysis. When $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$, the six kinds of objective function value obtained by the algorithm are shown in Figures 1 and 5 in both different scenarios. From Figure 1, we can see that the value of objective function obtained by proposed algorithm is smaller than those obtained by the compared algorithms with the number of VONs varying from 10 to 50. Similarly, it can be seen from the figure that when the number of VONs is from 3 to 7, the algorithm obtains an objective function value that is less than the objective function value obtained by the comparison algorithm. CAN-A considered four types of constraint nodes and links, wherein the substrate is configured subset of candidate paths and a subset of the candidate nodes in the substrate prior to embedding. GRC-SVNE maps selecting section capacity for all nodes as candidate nodes according to a node. The second stage uses the Dijkstra algorithm downlink map. This can lead to connection requests on a different link imbalance. The algorithm will get a maximum index with greater use of frequency bins. LCS D maximum link bandwidth requirements are mapped to link the shortest distance to reduce network costs, so that the connection request imbalances different links. IARA can search the optimal virtual nodes mapping scheme for all the virtual nodes and optimal routing scheme for all the connection requests using a search operators. In the first experiment scenario, when the number of virtual optical networks is 10, the ratio of the algorithm and the comparison algorithm to obtain the objective function value is 3.9%–7.4%. When the number of VONs is 50, ratio of the algorithm and the comparison algorithm to obtain the objective function value is 5.8%–13.1%. In addition, we can see that, with the increase in the number of VONs, the proposed algorithm can save more frequency slots. In a second experiment scenario, as shown in Figure 5, when the number of virtual nodes is 3, ratio of the algorithm and the comparison algorithm to obtain the objective function value is

4.5%–8.1%. When the number of virtual nodes is 7, ratio of the algorithm and five kinds of the comparison algorithm to obtain the objective function value is 5.2%–10.5%. When $\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$, the goal is to minimize the use of the maximum frequency bin index. Thus, the algorithm in two different experimental scenarios can save more frequency bins.

As can be seen from Figures 2 and 6, the objective function value obtained by the algorithm is less than the objective function value obtained by the comparison algorithm. The algorithm can determine the optimal objective function value with the smallest virtual node mapping and routing scheme. Further, the modulation level may be selected to minimize the objective function value. Thus, the algorithm's objective function value is minimal compared to those of the five kinds of the comparison algorithm. When the number of VONs is 10, the value of objective function by the proposed algorithm is 3.2%–9.8% less than that by the compared algorithms as shown in Figure 2. When the number of the virtual optical networks is 50, the value of the objective function of the algorithm, 5.8%, is smaller than the objective function value of the comparison algorithm, 12.5%. In Figure 6, when the number of nodes in each VON is 3, the value of objective function by the proposed algorithm is 5.1%–11.2%. When the number of nodes in each VON is 7, the value of the objective function of the algorithm is 6.8%–14.3% less than the objective function value of the comparison algorithm. When $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0$, the goal is to minimize the energy consumption (EC). Therefore, the algorithm can save more energy at two different experimental scenarios.

Similarly, when $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$, the target is $1 - f_3$. That is, the smaller the value of f is, the higher RFSU (frequency slots utilization) is. In Figures 3 and 7, the comparison frequency channel utilization algorithm obtained six kinds of three networks. The proposed algorithm evolution strategy, more balanced in the connection request K candidate path. Therefore, the algorithm can get a higher frequency than the other five algorithms in slot utilization.

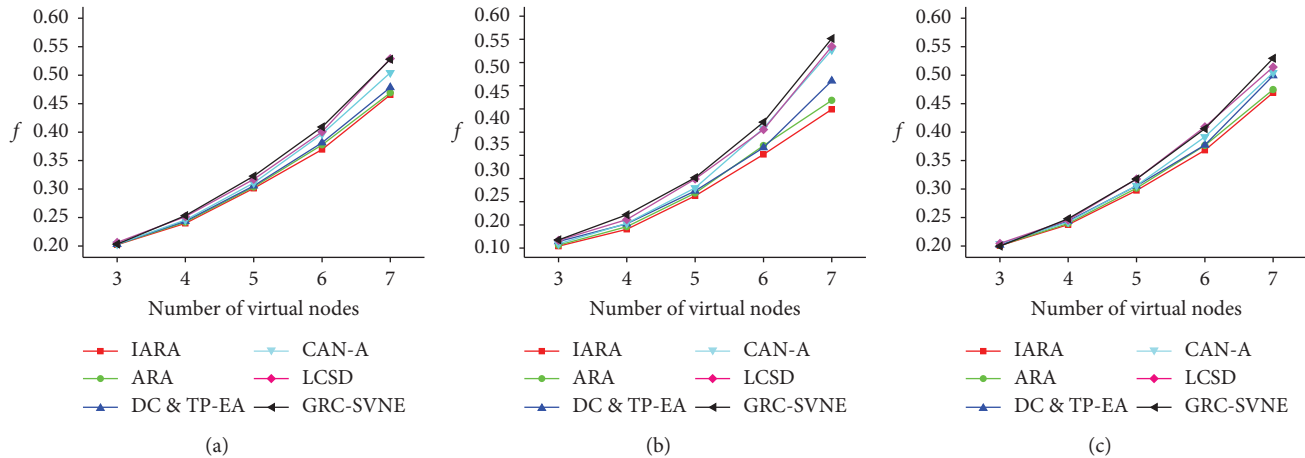


FIGURE 8: Experimental results in the second scene when $\alpha_1 = 1/3, \alpha_2 = 1/3, \alpha_3 = 1/3$. (a) Experimental results in NSFNET. (b) Experimental results in CHNNET. (c) Experimental results in ARPANET.

In Figure 3, in the three networks, when the number of VONs is from 10 to 50, the utilization of the algorithm frequency bins obtained is 42.1 to 49.5%, 35.8% to 48.1%, and 42.6% to 52.4%. In Figure 7, in the three networks, when the number of virtual optical network nodes is from 3 to 7, the utilization of the algorithm frequency bins obtained was 41.8% to 48.2%, 38.7% to 50.4%, and 44.1% to 56.3%. The results showed that, compared with other algorithms, it can improve the utilization rate of frequency bins.

Figures 4 and 8 give the three types in $\alpha_1 = 1/3, \alpha_2 = 1/3, \alpha_3 = 1/3$ comparison of the results of six kinds of experimental algorithm on the network. As can be seen, the objective function value obtained by the algorithm is less than the objective function value obtained by the comparison algorithm. The algorithm can determine the minimum objective function optimal virtual node mapping and routing scheme. As shown in Figure 4, when the number of virtual optical networks is 10, the objective function value obtained by the algorithm in this paper is 4.8%–10.2% smaller than the objective function value obtained by the comparison algorithm. When the number of the virtual optical networks is 50, the objective function value obtained by the algorithm in this paper is 6.7%–14.3% smaller than the objective function value obtained by the comparison algorithm. As shown in Figure 8, when the number of nodes in each virtual optical network is 3, the objective function value obtained by the algorithm in this paper is 5.2%–13.7% less than the objective function value obtained by the comparison algorithm. When the number of nodes in each virtual optical network is 7, the objective function value obtained by the algorithm in this paper is 7.1%–15.9% smaller than the objective function value obtained by the comparison algorithm.

5. Conclusion

We studied VONs mapping problems since generations. We established a maximum frequency bin index and studied the energy consumption of three-minimal-objective model accounting for frequency bin and determined optimum

solution for all connection requests virtual node mapping, routing, and spectrum allocation. In order to effectively solve three-objective optimization model, the weighted sum of the three strategies integrated into a target goal was minimized. Based on the idea of particle swarm optimization and differential evolution, an improved artificial raindrop algorithm (IARA) to effectively solve the biggest model was proposed. In comparison to other algorithms, for the simulation experiments under different test scenarios, test results show the effectiveness of the model and the algorithm proposed. However, the algorithm's complexity is high, only for static (offline) VONs problem.

Data Availability

All the data are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 31872704, 62006205, and 62002307), Innovation Team Support Plan of Henan University of Science and Technology (no. 19IRTSTHN014), Foundation of Henan Educational Committee (Contract no. 21A520039), and Nanhu Scholars Program for Young Scholars of XYNU, Youth Sustentation Fund of Xinyang Normal University (no. 2019-QN-040).

References

- [1] X. Xue and J. Chen, "Using compact evolutionary tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.
- [2] X. Li, T. Gao, L. Zhang, Y. Tang, Y. Zhang, and S. Huang, "Survivable k-node (edge) content connected virtual optical network (kc-von) embedding over elastic optical data center networks," *IEEE Access*, vol. 6, pp. 38780–38793, 2018.

- [3] H. Cui, S. Tang, F. Sun, Y. Xu, and X. Yang, "Topological embedding feature based resource allocation in network virtualization," *Mathematical Problems in Engineering*, vol. 2014, Article ID 271493, 10 pages, 2014.
- [4] J. Zhao, S. Subramaniam, and M. Brandt-Pearce, "Virtual topology mapping in elastic optical networks," in *Proceedings of the 2013 IEEE International Conference on Communications (ICC)*, pp. 3904–3908, IEEE, Budapest, Hungary, June 2013.
- [5] H. Li, L. Wang, X. Wen, Z. Lu, and J. Li, "MSV: an algorithm for coordinated resource allocation in network function virtualization," *IEEE Access*, vol. 6, pp. 76876–76888, 2018.
- [6] Z. Ding, Z. Xu, X. Zeng, T. Ma, and F. Yang, "Hybrid routing and spectrum assignment algorithms based on distance-adaptation combined coevolution and heuristics in elastic optical networks," *Optical Engineering*, vol. 53, no. 4, Article ID 046105, 2014.
- [7] M. Kalil, M. Youssef, A. Shami, A. Al-Dweik, and S. Ali, "Wireless resource virtualization: opportunities, challenges, and solutions," *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2690–2699, 2016.
- [8] Z. Li, W. Fei, and Z. Hongli, "QOS-aware resource allocation for network virtualization in an integrated train ground communication system," *Wireless Communications & Mobile Computing*, vol. 2018, Article ID 2653405, 12 pages, 2018.
- [9] B. Chen, J. Zhang, W. Xie, J. P. Jue, Y. Zhao, and G. Shen, "Cost-effective survivable virtual optical network mapping in flexible bandwidth optical networks," *Journal of Lightwave Technology*, vol. 34, no. 10, pp. 2398–2412, 2016.
- [10] J. F. Botero and X. Hesselbach, "Greener networking in a network virtualization environment," *Computer Networks*, vol. 57, no. 9, pp. 2021–2039, 2013.
- [11] A. Khan, A. Zugenmaier, D. Jurca, and W. Kellerer, "Network virtualization: a hypervisor for the internet?" *IEEE Communications Magazine*, vol. 50, no. 1, pp. 136–143, 2012.
- [12] A. Berl, N. Race, J. Ishmael, and H. De Meer, "Network virtualization in energy-efficient office environments," *Computer Networks*, vol. 54, no. 16, pp. 2856–2868, 2010.
- [13] R. Lu and X. Nan, "Survivable multipath routing and spectrum allocation in OFDM-based flexible optical networks," *Journal of Optical Communication and Network*, vol. 5, no. 3, pp. 172–182, 2013.
- [14] P. Zhang, H. Li, Y. Ni, F. Gong, M. Li, and F. Wang, "Security aware virtual network embedding algorithm using information entropy topsis," *Journal of Network and Systems Management*, vol. 28, no. 1, pp. 35–57, 2020.
- [15] B. Farkiani, B. Bakhshi, and S. A. MirHassani, "Stochastic virtual network embedding via accelerated benders decomposition," *Future Generation Computer Systems*, vol. 94, pp. 199–213, 2019.
- [16] P. Zhang, H. Yao, M. Li, and Y. Liu, "Virtual network embedding based on modified genetic algorithm," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 1–12, 2017.
- [17] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding," *ACM Sigcomm Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.
- [18] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.
- [19] E. Amaldi, S. Coniglio, A. M. C. A. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electronic Notes in Discrete Mathematics*, vol. 52, pp. 213–220, 2016.
- [20] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding LC-VNE algorithms towards integrated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3648–3661, 2016.
- [21] J. Liao, M. Feng, S. Qing, T. Li, and J. Wang, "Live: learning and inference for virtual network embedding," *Journal of Network and Systems Management*, vol. 24, no. 2, pp. 227–256, 2016.
- [22] Z.-P. Cai, Q. Liu, P. Lü, and Z.-Y. Wang, "Virtual network mapping model and optimization algorithms," *Journal of Software*, vol. 23, no. 4, pp. 864–877, 2012.
- [23] M. He, L. Zhuang, S. Tian, G. Wang, and K. Zhang, "DROI: energy-efficient virtual network embedding algorithm based on dynamic regions of interest," *Computer Networks*, vol. 166, p. 106952, 2020.
- [24] T. Huang, Y. Gu, J. Liu, and Y. Liu, "Time efficient virtual network embedding algorithm," *Intelligent Automation & Soft Computing*, vol. 22, no. 2, pp. 273–280, 2016.
- [25] H. Liu, J. Du, Y. Chen, M. Xiang, and Y. Ma, "A coordinated virtual optical network embedding algorithm based on resources availability-aware over elastic optical networks," *Optical Fiber Technology*, vol. 45, pp. 391–398, 2018.
- [26] M. Pourvali, H. Bai, J. Crichigno, and N. Ghani, "Multicast virtual network services embedding for improved disaster recovery support," *IEEE Communications Letters*, vol. 22, no. 7, pp. 1362–1365, 2018.
- [27] X. Liu, Z. Zhang, J. Li, and S. Su, "Clustering-based energy-aware virtual network embedding," *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, Article ID 15501477, 2017.
- [28] H. Xuan, S. Wei, Y. Feng, H. Guo, and Y. Li, "A new bi-level mathematical model and algorithm for VONs mapping problem," *IEEE Access*, vol. 8, pp. 101797–101811, 2020.
- [29] A. Bocoi, S. Matthias, R. Franz, K. Moritz, B. Christian-Alexander, and S. Bernhard, "Reach-dependent capacity in optical networks enabled by OFDM," in *Proceedings of the OFC (OMQ4)*, pp. 1–3, San Diego, CA, USA, March 2009.
- [30] M. Zhu, Q. Sun, S. Zhang, P. Gao, B. Chen, and J. Gu, "Energy-aware virtual optical network embedding in sliceable-transponder-enabled elastic optical networks," *IEEE Access*, vol. 7, pp. 41897–41912, 2019.
- [31] H. Cao, Y. Zhu, G. Zheng, and L. Yang, "A novel optimal mapping algorithm with less computational complexity for virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 356–371, 2018.