

# A Virtual Network Embedding Algorithm Based on Cellular Automata Genetic Mechanism

Lei Zhuang, Guoqing Wang<sup>a</sup>, Ming Wang and Kunli Zhang

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

**Abstract.** The optimal embedding problem of virtual network requests, which satisfies nodes and link constraints, is a NP-hard problem. Heuristic algorithms solve the problem with the mathematical model optimization, but it fails to consider the influence of the virtual network embedding node itself on the optimal solution. So the cellular automata genetic mechanism is introduced into the problem, and the virtual network embedding algorithm based on cellular genetic algorithm (VNE-CGA) has been proposed. VNE-CGA uses the cellular automata to model the node, and replaces the "B4567/S1234" rule with the crossover operation in genetic algorithm. Through learning from neighbours to guide the individual's optimization process, VNE-CGA improves the inherent defects of traditional genetic algorithm. The experimental results show that the request acceptance ratio and the long-term average revenue increase about 5% and 12%.

## 1 Introduction

With the expansion of the Internet and the growth of new services, the traditional network structure tends to be "rigid", and network virtualization provides strong support for the innovation and sustainable development of the network [1, 2]. The core idea of the network virtualization is constructing its own virtual network (VN) for different services on the same physical network [3]. It is a NP-hard problem to provide the underlying physical network resources to virtual network requests on demand [4]. Heuristic algorithms are the main means to solve this kind of problem. Ref. [5] proposed a load balancing algorithm based on particle swarm optimization to improve long-term average revenue of the underlying network. Ref. [6] proposed a virtual network embedding algorithm based on ant colony algorithm, which uses ant colony iteration and intelligence to solve this optimization problem. Ref. [7] first used genetic algorithm (GA) to solve the embedding optimization problem, but the convergence speed was slow and easily fell into the local optimization. Ref. [8] proposed an iterative optimization embedding solution VNE-PSO-GA, which merges genetic algorithm and particle swarm optimization algorithm together, to solve the local optimal problem. Ref. [9] improved the genetic algorithm with the simple algorithm (VNE-M-GA), which solves the premature convergence problem to a certain extent.

Most of the above studies are solved and optimized from the perspective of mathematical models, and less consideration is given to the impact of the changes of nodes and links in the virtual network embedding (VNE) on the final results. Cellular automata (CA) rely on simple local rules to study the overall complexity and are widely used in fields such as information transfer [10], image

encryption [11], complex function optimization [12] and biological simulation [13]. Especially for many single-objective optimization problems, cellular genetic algorithm has achieved good results as a mature and efficient solution [14, 15]. Based on this, cellular automata are used to model physical nodes and links, and cellular genetic algorithm is used to solve the problem of virtual network embedding. A virtual network embedding algorithm based on cellular automata genetic mechanism (VNE-CGA) is proposed. According to the searched references, it is a new method to consider with cellular automata into the virtual network embedding problem. Compared with the traditional genetic algorithm and the improved algorithm proposed in Ref. [8, 9], VNE-CGA improves the request acceptance ratio and the long-term average revenue of virtual network.

Section 2 introduces the basic concepts of virtual network embedding, puts forward the evaluation criterion and establishes the mathematical model. Section 3 studies the genetic algorithm in virtual network embedding to analyze its deficiencies, and then proposes VNE-CGA using cellular automata to improve the genetic algorithm. Section 4 introduces the experimental environment and analyzes experimental results. Finally, section 5 gives the conclusion and provides ideas for future researches.

## 2 Virtual network embedding problem

### 2.1 Problem description

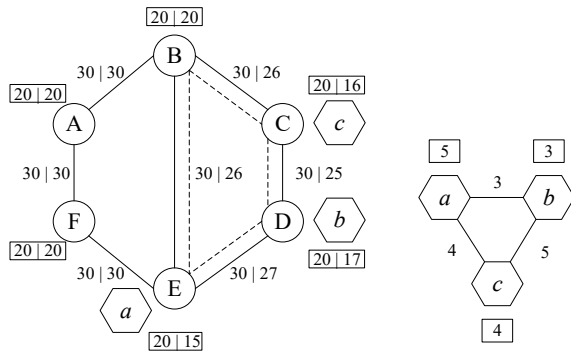
1) The underlying physical network:  $G_s(N_s, L_s)$  means undirected graph with weight, where  $N_s$  is the set of underlying physical network nodes and  $L_s$  is the set of

<sup>a</sup> Corresponding author: [iegqwang@163.com](mailto:iegqwang@163.com)

underlying physical network links.  $c(n_s)$  denotes the remaining CPU resource, where  $n_s \in N_s$ .  $b(l_s)$  denotes the available bandwidth resource, where  $l_s \in L_s$ . Figure 1 (i) displays the underlying physical network, which has 6 nodes and 7 links. The number on the left of the rectangle represents the  $c(n_s)$  before embedding, and the right number represents the  $c(n_s)$  when embedding completed. The two numbers nearby the link represent the  $b(l_s)$ , which have the same conditions like  $c(n_s)$ .

2) The virtual network request:  $G_v(N_v, L_v)$  means undirected graph with weight, where  $N_v$  is the set of virtual network nodes and  $L_v$  is the set of virtual network links.  $c(n_v)$  denotes the required CPU resource, where  $n_v \in N_v$ .  $b(l_v)$  denotes the required bandwidth resource, where  $l_v \in L_v$ . Figure 1 (ii) displays the virtual network request, which has 3 virtual nodes and 3 virtual links. For example, the required CPU resource in virtual node  $a$  is 5, and the required bandwidth resource on virtual link  $(a, b)$  is 3.

3) The virtual network embedding:  $M: G_v(N_v, L_v) \rightarrow G_s(N_s, L_s)$  is a mapping, where  $N_{s'} \in N_s$  and  $L_{s'} \in L_s$ .



(i) The underlying physical network (ii) The VN request

**Figure 1.** Virtual network embedding problem

## 2.2 Evaluation criterion

The primary research is that cellular genetic algorithm was used to solve the virtual network embedding optimization problem with nodes and links constraints. The goal is to reduce the operating costs and improve the operating profit of the physical network. According to Ref. [9], two definitions about the revenue and the cost were summarized.

The revenue of an accepted virtual network request at time  $t$  is defined as the sum of the resources it needs. The formal definition is as follows:

$$R(G_v, t) = (1 - \alpha) \sum_{n_v \in N_v} c(n_v) + \alpha \sum_{l_v \in L_v} b(l_v) \quad (1)$$

where  $\alpha$  denotes the relative weight of the link resources and CPU resources. The value of  $\alpha$  is between 0 and 1.  $\alpha$  is closer to 1, means the revenue generated by unit link resource is greater, otherwise means the revenue generated by unit CPU resource is greater.

The cost of an accepted virtual network request at time  $t$  is defined as the sum of the resources it consumed. The formal definition is as follows:

$$C(G_v, t) = (1 - \alpha) \sum_{n_v \in N_v} c(n_v) + \alpha \sum_{l_v \in L_v} \sum_{l_s \in L_s} b(f_{l_s}^{l_v}, l_v) \quad (2)$$

where  $f_{l_s}^{l_v} \in \{0,1\}$ . When the physical link  $l_s$  allocates bandwidth resource to the virtual link  $l_v$ , the value of  $f_{l_s}^{l_v}$  is 1, otherwise is 0.  $b(f_{l_s}^{l_v}, l_v)$  denotes the value of the assigned bandwidth from  $l_s$  to  $l_v$ . The meaning of  $\alpha$  is same with (1).

Based on two definitions, there are three main evaluation criterions of the virtual network embedding.

1) The long-term average revenue of underlying physical network defines as:

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G_v, t)}{T} \quad (3)$$

2) The request acceptance ratio of virtual network defines as:

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T VNR_s}{\sum_{t=0}^T VNR} \quad (4)$$

where  $\sum_{t=0}^T VNR_s$  denotes the number of virtual network requests, who embedded successfully, from time 0 to  $T$ , and  $\sum_{t=0}^T VNR$  denotes the total number of requests from time 0 to  $T$ .

3) The average revenue/cost ratio defines as:

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

## 2.3 Integer linear programming model

The optimization problem of virtual network embedding aims to improve the profitability of the underlying network operation, which needs to establish the corresponding model of mixed integer linear programming. The model involves two variables.

Binary variable  $f_{uv}^{ij}$ : The value of 1 indicates the physical link  $(i, j)$  allocates bandwidth resource to the virtual link  $(u, v)$ , and 0 means it is unallocated.

Binary variable  $x_u^i$ : The value of 1 indicates the virtual node  $u$  is embedded on the physical node  $i$ , and 0 means there is no embedding.

For a virtual network embedding request, the cost of node embedding is constant, and the cost of bandwidth allocation varies from different scenarios. Therefore, the objective function of the model should be proposed as:

$$\text{Minimize} \sum_{(u,v) \in L_v} \sum_{(i,j) \in L_s} f_{uv}^{ij} \times b(l_{uv}) \quad (6)$$

If the VNE needs to be completed once, (6) must meet the capacity constraints, that means physical nodes and links must have the sufficient resources for embedding successfully. The available computing resources of the

physical nodes must meet the computing resources requirements of the virtual nodes, such that:

$$\begin{cases} \forall u \in N_v, \forall i \in N_s \\ x_u^i \times c(u) \leq c(i) \end{cases} \quad (7)$$

The available bandwidth resources for the physical links must satisfy the bandwidth requirements for the virtual links, such that:

$$\begin{cases} \forall (i, j) \in L_s, \forall (u, v) \in L_v \\ f_{uv}^{ij} \times b(l_{uv}) \leq b(l_{ij}) \end{cases} \quad (8)$$

In one virtual network request, each underlying physical node can only be divided into one virtual node, and one virtual node can only be embedded to one physical node. The constraints are as follows:

$$\begin{cases} \forall w \in N_s, \sum_{m \in N_v} x_{mw} \leq 1 \\ \forall m \in N_v, \sum_{w \in N_s} x_{mw} \leq 1 \end{cases} \quad (9)$$

If the virtual node  $u$  and  $v$  are embedded on the physical node  $i$  and  $j$  separately, the virtual link  $(u, v)$  will be embedded on the underlying physical link  $P$  from node  $i$  to  $j$ . At the source node  $i$ , the outgoing flow is 1 and the incoming flow is 0, so  $\sum_{(i,j) \in L_s} f_{uv}^{ij} - \sum_{(j,i) \in L_s} f_{uv}^{ji} = 1$ . At the meeting node  $j$ , the outgoing flow is 0 and the incoming flow is 1, so  $\sum_{(i,j) \in L_s} f_{uv}^{ij} - \sum_{(j,i) \in L_s} f_{uv}^{ji} = -1$ . At the other node on link  $P$ , the outgoing and incoming flow are 1, so  $\sum_{(i,j) \in L_s} f_{uv}^{ij} - \sum_{(j,i) \in L_s} f_{uv}^{ji} = 0$ . The connectivity constraints are as follows:  $\forall i \in N_s, \forall (u, v) \in L_v$ ,

$$\sum_{(i,j) \in L_s} f_{uv}^{ij} - \sum_{(j,i) \in L_s} f_{uv}^{ji} = \begin{cases} 1, & \text{if } x_u^i = 1 \\ -1, & \text{if } x_u^j = 1 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

### 3 VNE-CGA

#### 3.1 Genetic algorithm in VNE

1) Coding and generating the initial population. The coding uses real numbers. Let  $N_v = \{n_v^1, n_v^2, \dots, n_v^L\}$ , where  $L = |N_v|$ , that means the total number of virtual nodes is  $L$ . The sequence with long  $L$  represents the solution (chromosome) of the problem. The value of  $n_v^i$  represents the  $i$ th physical nodes which the virtual node has been embedded on. The initial population is produced by generating  $N$  chromosomes randomly.

2) Crossover operator. In the traditional genetic algorithm, the partial mapping crossover method is often used. First, the population is paired with the random method. Then the location of the intersection is determined randomly. Finally, it is on probability  $P_c$  to exchange part of genes between the paired chromosomes.

3) Mutation operator. The random method is used to set the genetic variation position of each individual, and

then change the value of the gene according to the mutation probability  $P_m$ .

4) Feasible solution examination. Solutions after crossover and mutation may not be viable to the problem. For example, let the parent strings  $A = (18,6|7,8,3|4,22,25)$  and  $B = (3,38|7,9,12|4,16,13)$ . At this point, the physical node 22 carries the 3rd and the 8th virtual nodes at the same time, which does not meet the constraint requirement in (9). Then the 8th virtual node needs to be embedded again using the random method until the constraint is satisfied. So  $A' = (18,6|22,8,3|35,22,25)$  after mating.

5) Fitness evaluation. For a given chromosome (i.e., an embedding scheme), the optimization problem of original resource allocation has been reduced to the problem of virtual network links embedding, that is multi-commodity flow (MCF) problem. In the proposed algorithm, the objective function has a constant nonnegative value and the optimization direction is to find the minimum value. Therefore, the value of (6) can be directly used as the fitness of the individual.

Through the description and research of traditional genetic algorithm, there are three problems as follows:

1) When the traditional genetic algorithm chooses the next generation of individuals, the possibility of individuals to be selected depends on the fitness, but the selection process is probabilistic. If fully comply with this basic model, an individual with higher fitness may be unselected in certain generations. On the contrary, if a "leader" (individual with high fitness) within the population is selected each time and some "potential shares" are ignored, the population is likely caught in local optimization.

2) For the problem of dynamic virtual network request, solving with traditional genetic algorithm may no longer react to the data that changes later due to precocity.

3) The crossover operation ensures that some possible solutions are not lost, but on the other hand, especially for the initial population, crossover operation has the same effect as setting a large mutation rate. In this process, crossover is not properly guided and learning between individuals is blind.

#### 3.2 Improvement with cellular automata

As described in Section 3.1, traditional genetic algorithm tends to converge prematurely and fall into local optimization. In order to improve its optimization ability and the performance of the algorithm, cellular automata are introduced. Through the analysis of the VNE, the CA were used to model network nodes. Combined with traditional genetic algorithm, an embedding algorithm was proposed that aimed to maximize the operating revenue of the underlying network.

CA are dynamic system, whose space and time are discrete. It is widely used in the research and analysis of complex problems. Its most basic components include cellular, space, neighbors and rules [16].

1) Cellular: In VNE-CGA, let the node embedding scheme  $N_v = \{n_v^1, n_v^2, \dots, n_v^L\}$  as an individual for coding. Node number  $n_v^L$  uses 8 bit (LB) binary number to indicate. The encoded individual is a LB-length binary vector. Each encoded bit is a cell.

2) Space: For the problem of VNE, an undirected graph with weighted is used to represent the underlying network and virtual network. Therefore, the 2-dimensional spatial structure is used to model. The chromosomes in the population are arranged in rows to form a (LB) $q$  binary matrix, where  $q$  is the number of chromosomes.

3) Neighbors: Statuses updating in cellular automata only depends on their own status and the statuses of their neighbors. Common neighbor models include Von Neumann, Moore and extended Moore neighbor model. Von Neumann neighbor model is simple to calculate, but the Moore neighbor model is more stable. And the extended Moore neighbor model needs a large amount of calculation. So VNE-CGA used Moore neighbor model to calculate and individual  $C_{ij}$  had 8 neighbors. They are  $C_{i-1,j-1}$ ,  $C_{i-1,j}$ ,  $C_{i-1,j+1}$ ,  $C_{i,j-1}$ ,  $C_{i,j+1}$ ,  $C_{i+1,j-1}$ ,  $C_{i+1,j}$  and  $C_{i+1,j+1}$ .

4) Rules: Rules equal the status transfer functions such that  $f: S_i^{t+1} = f(S_i^t, S_N^t)$ , where  $S_N^t$  denotes the set of neighbors of cell  $i$  at time  $t$ . The classic "Life Game (B3/S23 rule)" is described as follows: If a dead cell has three living neighbors, its propagation changes the status from 0 to 1, otherwise the status remains; If a live cell has two or three living neighbors, its survival makes the status unchanged, otherwise the status changes from 1 to 0. The expression of evolution rule is as follows:

$$\begin{cases} \text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1, & \text{if } \sum S_N^t = 3 \\ 0, & \text{if } \sum S_N^t \neq 3 \end{cases} \\ \text{if } S_i^t = 1 \text{ then } S_i^{t+1} = \begin{cases} 1, & \text{if } \sum S_N^t = 2, 3 \\ 0, & \text{if } \sum S_N^t \neq 2, 3 \end{cases} \end{cases} \quad (11)$$

As the virtual network embedding progresses, the underlying physical resources change from moment to moment. Especially when the underlying network is overloaded, the traditional "best effort" mode is likely to cause unbalanced utilization of the underlying resources, which leads to loss of more future benefits [17]. In order to improve the profitability of the underlying network providers, it is necessary to consider minimizing the cost of using the underlying resources in the process of VNE, as well as balancing utilization of the underlying resources. Combined with the classic evolution rule "Life Game (B3/S23 rule)", the recovery and distribution of the physical resources are considered as the extinction and multiplication of living things. Through the experimental analysis, the evolution rule "B4567/S1234" was established to improve VNE-CGA. The expression of evolution rule is as follows:

$$\begin{cases} \text{if } S_i^t = 0 \text{ then } S_i^{t+1} = \begin{cases} 1, & \text{if } \sum S_N^t = 4,5,6,7 \\ 0, & \text{if } \sum S_N^t \neq 4,5,6,7 \end{cases} \\ \text{if } S_i^t = 1 \text{ then } S_i^{t+1} = \begin{cases} 1, & \text{if } \sum S_N^t = 1,2,3,4 \\ 0, & \text{if } \sum S_N^t \neq 1,2,3,4 \end{cases} \end{cases} \quad (12)$$

### 3.3 Algorithm description

VNE-CGA uses evolution rule B4567/S1234 to replace the crossover between genomes in traditional genetic algorithm. In VNE-CGA, the crossover operation no longer limited to matching two chromosome chains, and each cell, which is considered as an individual's gene, needs to scan when it is not on the boundary. The status of the neighbor model determines next status at the next moment. The crossover operation is shown in Figure 2, where row means individual and column means gene.

	G1	G2	G3	G4	G5	G6	G7	G8	G9
R1									
R2									
R3									
R4									

(i) Chromosome chains before crossover

	G1	G2	G3	G4	G5	G6	G7	G8	G9
R1									
R2									
R3									
R4									

(ii) Chromosome chains after crossover

**Figure 2.** Crossover operators in VNE-CGA

In figure 2, white represents the status of 0 and the black represents the status of 1. As shown in Figure 2 (i), the gene G2 of chromosome R2 was white, whose eight neighbors had 4 white and 4 black. So that  $S_i^t = 0$ ,  $\sum S_N^t = 4$ , and  $S_i^{t+1} = 1$  was displayed in Figure 2 (ii).

VNE-CGA algorithm is described as follows:

**Input.** PN  $G_s = (N_s, E_s)$ , VN request  $G_v = (N_v, E_v)$

**Output.** VNE scheme *Result*

**Procedure.**

- Step 1: Accept arrived virtual network request
- Step 2: Generate the initial population  
The population  $P(0)$  initialize randomly  
The size is  $N$  and the max of iterations is  $T$   
 $P_c$  denotes crossover probability  
 $P_m$  denotes mutation probability  
 $fit_{max} = 0, M_{best} = \emptyset, B_{best} = \emptyset, Result = \emptyset$
- Step 3: Calculate the fitness of each individual  
for each chromosome  $chr \in A$  do  
uses ILOG CPLEX to solve (6)-(10)  
obtains the fitness  $fit(chr)$   
gets the corresponding supply solutions  
if  $fit(chr) > fit_{max}$   
 $fit_{max} = fit(chr)$   
 $M_{best} \leftarrow chr$   
 $B_{best} \leftarrow b(chr)$   
end if  
end for
- Step 4: Generate the next population  
According to fitness,  $N$  chromosomes are selected  
All of genes make up the cellular space  
The neighbor matrix is created with Moore model  
It is on probability  $P_c$  to follow B4567/S1234 rule  
For new population mutation, the probability is  $P_m$   
The feasible solutions examine the errors  
Population  $P(t)$  evolve into  $P(t + 1)$

- Step 5: Check the fitness or iteration times  
 if  $t = T$  or the fitness change ratio does not exceed  $q$  continuous  $n$  times

$$Result = M_{best} \cup B_{best}$$

else

jump to Step 3

end if

- Step 6: Algorithm end

The time complexity of genetic algorithm is  $O(MN^3)$  [18], where  $M$  denotes the number of goals and  $N$  denotes the number of individuals. The time complexity of single-objective genetic algorithm is  $O(N^3)$ . VNE-CGA replaces the single point or multi-point crossover with cellular rules to evolve, mainly in the process of the next generation population (Step 4). Although the calculation has increased by improvement, the time complexity of VNE-CGA is still  $O(N^3)$ .

## 4 Experimental results

Due to the integration of cellular automata and genetic algorithms, VNE-CGA can solve the problem of virtual network embedding. Therefore, compared with the traditional GA, VNE-PSO-GA [8] and VNE-M-GA [9], the performance of the VNE-CGA in solving the problem of VNE is analyzed.

The experimental environment and parameters are similar with Ref. [8] and [9]. The network topology is randomly generated by using GT-ITM tools, including 100 physical nodes and about 500 physical links. The available resources obey the uniform distribution of [50-100]. Each simulation experiment runs about 50,000 time units and contains 2,500 virtual network requests, where the requests in 100 time units follow the Poisson distribution of mean 5 and the life cycle obeys an exponential distribution. The virtual nodes in each virtual network request connect on probability 0.5, and the number of nodes and resource requirements satisfy the uniform distribution of [2-20] and [0-50] respectively. Initial population size  $N$  sets to 16, the maximum number of iterations  $T$  sets to 10, the crossover probability  $P_c$  sets to 0.5, and the mutation probability  $P_m$  sets to 0.01.

The long-term average revenue of underlying physical network and the request acceptance ratio of virtual network are respectively shown in Figure 3 and 4. The horizontal axis represents the running time, and the unit is 1000 time units. Compared with GA, VNE-PSO-GA and VNE-M-GA, VNE-CGA obtains the better results. The long-term average revenues have increased by 26%, 12% and 6%, and the request acceptance ratios have increased by 8%, 5% and 3%. The main reason is that GA and VNE-PSO-GA are easy to get local optimization, and VNE-M-GA with simplex method makes the convergence performance decrease. According to the characteristics of VNE itself, the cellular automata model is introduced to make the interaction between individuals Omnidirectional and dynamic, which ensures the diversity among the populations and effectively improves the local optimization shortcoming to avoid premature

convergence. While reducing the operating costs of the underlying network, it accepts more virtual network requests and increases long-term revenue.

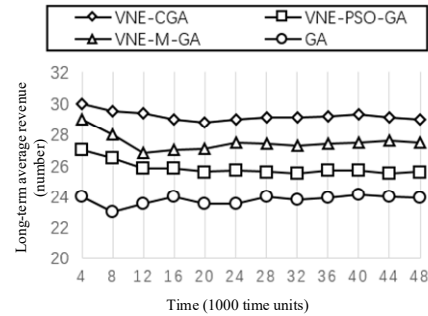


Figure 3. Long-term average revenue

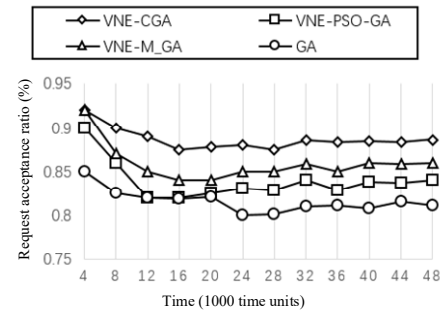


Figure 4. Request acceptance ratio of VN

The average revenue/cost ratio is shown in Figure 5. Compared with GA, VNE-PSO-GA and VNE-M-GA, VNE-CGA improves the ratio by 27%, 10%, and 4%. According to the above analysis, VNE-CGA has greatly improved than the traditional GA in terms of the long-term revenue of underlying network and the acceptance ratio of virtual network requests. Compared with the improved VNE-PSO-GA and VNE-M-GA, it also has a certain degree of improvement.

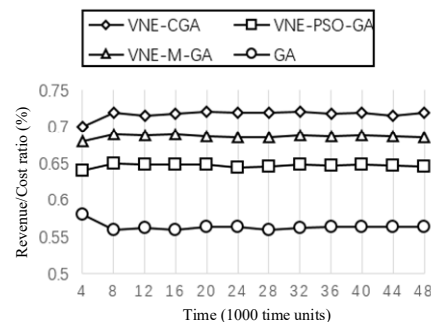
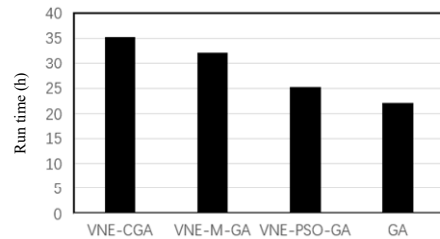


Figure 5. The average revenue/cost ratio

The run time of the algorithm is shown in Figure 6. The runtime of VNE-CGA, VNE-PSO-GA and VNE-M-GA have increased from the traditional GA. Because the algorithms are based on GA to improve, that results in the computational complexity of the algorithm increased. VNE-CGA, however, consumes more time than other algorithms because the more extensive cellular space and the limited range of neighbors also determine that. It takes a certain amount of time for optimal information to be transmitted in the cellular space, which indirectly affects convergence speed. In the large-scale real application environment, the computing ability of the hardware is much better than the simulation experiment environment.

The cellular automata are more convenient to carry on the parallel operation [19], and the calculation time difference will be further reduced. In summary, the actual benefits of the VNE-CGA are much higher than the computational costs of the algorithm.



**Figure 6.** Algorithm run time

## 5 Conclusion

In order to maximize the long-term average revenue of the underlying physical network, an integer linear programming model was established to optimize the virtual network embedding. Based on the characteristics of the virtual network embedding, VNE-CGA was proposed to solve the embedding optimization problem. The algorithm uses the evolution rules of cellular automata to redefine the crossover operation, which improves the searching ability of the algorithm. The experimental results show that VNE-CGA improves the premature convergence to a certain degree, reduces the overhead of the underlying physical network, and enhances the request acceptance ratio and the long-term average revenue. In large-scale practical applications, the virtual network embedding is more complex. The adaptive dynamic neighbor model should be studied in future, so that VNE-CGA can be suitable for different scenarios.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grant No.61379079, the Science and Technology Key Project of Henan under Grant No.172102210478, and the Key Scientific Research Project of Higher Education of Henan under Grant No.17A520057.

## References

1. J. Inführ, G. Raidl: A memetic algorithm for the virtual network mapping problem. *J. Heuristics* **22**(4), 475-505 (2016).
2. A. Fischer, J. F. Botero, M. T. Beck, H. D. Meer, X. Hesselbach: Virtual network embedding: a survey. *IEEE Commun. Surv. Tut.* **15**(4), 1888-1906 (2013).
3. N. M. M. K. Chowdhury, R. Boutaba: A survey of network virtualization. *Comput. Netw.* **54**(5), 862-876 (2010).
4. E. Amaldi, S. Coniglio, A. M. C. A. Koster, M. Tieves: On the computational complexity of the

- virtual network embedding problem. *Electron. Notes Discret. Math.* **52**, 213-220 (2016).
5. C. Wang, Y. Su, L. Zhou, S. Peng, Y. Yuan, H. Huang: A virtual network embedding algorithm based on hybrid particle swarm optimization. In: *International Conference on Smart Computing and Communication*, pp. 568-576. Springer, Heidelberg (2016).
6. X. Liu, Z. Zhang, X. Li, S. Su: Optimal virtual network embedding based on artificial bee colony. *EURASIP J. Wirel. Comm.* **2016**(1), 273(2016).
7. G. Sun: Virtual network embedding technology research. Doctoral dissertation, University of Electronic Science and Technology of China. Chengdu Sichuan, P.R.China (2012).
8. W. Wang, B. Wang, Z. Wang, C. Xing: Virtual network embedding algorithm based on a hybrid swarm intelligence optimization. *J. Comput. Appl.* **34**(4), 930-934 (2014).
9. J. Liu, T. Song, Y. Hu, L. Zhuang: Research on virtual network mapping based on mixed genetic algorithm. *J. Chin. Comput. Syst.* **37**(4), 773-777 (2016).
10. B. Cissé, S. E. Yacoubi, S. Gourbière: The basic reproduction number for Chagas disease transmission using cellular automata. In: *International Conference on Cellular Automata*, pp. 278-287. Springer, Heidelberg (2014).
11. A. Bakhshandeh, Z. Eslami: An authenticated image encryption scheme based on chaotic maps and memory cellular automata. *Opt. Laser. Eng.* **51**(6), 665-673 (2013).
12. M. Alabbas, S. Jaf, A. H. M. Abdullah: Optimize BPNN using new breeder genetic algorithm. In: *International Conference on Advanced Intelligent Systems and Informatics*, pp. 373-382. Springer, Heidelberg (2016).
13. J. Was, G. C. Sirakoulis: Special issue on simulation with cellular automata. *Simul. T. Soc. Mod. Sim.* **92**(2), 99-100 (2016).
14. E. Alba, B. Dorronsoro: The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE T. Evolut. Comput.* **9**(2), 126-142 (2005).
15. S. Karthikeyan, M. Saravanan, M. Rajkumar: Optimization of worker assignment in dynamic cellular manufacturing system using genetic algorithm. *J. Manuf. Syst.* **15**(1), 35-42 (2016).
16. S. Wolfram: Cellular automata as models of complexity. *Nature* **311**(5985), 419-424 (1984).
17. J. Yu, C. Wu: Randomized algorithm for virtual network mapping problem based on load balancing. *Comput. Sci.* **41**(6), 69-74 (2014).
18. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T. Evolut. Comput.* **6**(2), 182-197 (2002).
19. G. C. Sirakoulis: Parallel application of hybrid DNA cellular automata for pseudorandom number generation. *J. Cell. Autom.* **11**(1), 63-89 (2016).