

Adaptive Network Slicing Control Method for Unpredictable Network Variations using Quality-Diversity Algorithms

Amato Otsuki Graduate School of Information Science and Technology, Osaka University Osaka, Japan a-otsuki@ist.osaka-u.ac.jp	Daichi Kominami Graduate School of Information Science and Technology, Osaka University Osaka, Japan d-kominami@ ist.osaka-u.ac.jp	Hideyuki Shimonishi Cybermedia Center, Osaka University shimonishi.cmc@ osaka-u.ac.jp	Masayuki Murata Graduate School of Information Science and Technology, Osaka University Osaka, Japan murata@ist.osaka-u.ac.jp	Tatsuya Otoshi Graduate School of Economics, Osaka University Osaka, Japan t-otoshi@econ.osaka-u.ac.jp
--	---	---	---	---

Abstract—Network slicing technology is required to dynamically provide virtual networks in response to user requirements with a wide variety of services operating on the network. Generally, optimal allocation of virtual networks to resources on the real network is a combinatorial optimization problem, and it is difficult to find an exact solution in realistic time in the current large-scale and complex networks. In addition, user requirements change dynamically, and therefore, optimization methods that can cope with such temporal variations in the situation are required. In this paper, we propose a method to solve a virtual network embedding problem using quality-diversity (QD) algorithms, especially the MAP-Elites algorithm, and evaluate its effectiveness through computer simulations.

Index Terms—Network virtualization, network slicing, SDN (Software Defined Networking), genetic algorithm, genetic diversity.

I. INTRODUCTION

In recent years, as cloud services and IoT technologies have become increasingly popular, there is a growing need to provide users with flexible networks through network virtualization [1], [2]. The construction of multiple virtual networks on a substrate network is called network slicing and is considered a fundamental technical specification in 5G networks. To control network slicing, it is necessary to determine the mapping between substrate and virtual networks, and the problem of finding the mapping is called Virtual Network Embedding (VNE) [3].

Since the computational complexity to find the optimal mapping increases exponentially with the size of the network, it is required to heuristically solve this problem in modern networks, which have become increasingly large and complex. Furthermore, since the structure of the virtual network dynamically changes according to user requirements in practical use, it is important to be able to derive a solution immediately in response to such changes.

One of the algorithms for handling such combinatorial optimization problems is an evolutionary algorithm that models the

mechanism of biological evolution. In the genetic algorithm, which is the most representative of these algorithms, optimization is performed by repeatedly selecting individuals with high evaluation values preferentially from multiple randomly generated individuals and applying genetic operations such as crossover and mutation for them. Due to its ability to handle optimization problems independently of the characteristics of the objective function, it has been applied in a wide range of fields. However, it is known that in such conventional genetic algorithms, populations tend to converge to a single form, unlike the natural world where diverse organisms can be found. Therefore, novel optimization methods have been proposed to generate a variety of solutions while maintaining diversity, and these methods are called quality-diversity (QD) algorithms [4]. They can improve their adaptability to diverse environments by simultaneously searching for superior solutions with different characteristics.

In this paper, we propose a method using QD algorithms, in particular, the MAP-Elites algorithm [5] to achieve rapid adaptation for dynamic VNE problems with unpredictable environmental variations. We show that maintaining solution diversity promotes optimization and adaptation to environmental variations through a comparison between conventional genetic algorithms and QD algorithms and heuristics.

The remainder of this paper is organized as follows. Section II formulates the VNE problem. Section III introduces MAP-Elites, which is one of the QD algorithms that we mainly focus on and details of our method. Section IV shows the results of computer simulations. Section V summarizes the paper.

II. PROBLEM FORMULATION

The virtual network embedding (VNE) problem is a resource allocation problem in network virtualization and involves substrate and virtual networks. A substrate network consists of multiple substrate nodes, such as servers and

switches, and substrate links that connect them, with finite substrate resources such as CPU and link bandwidth. A virtual network consists of virtual nodes running on substrate nodes and virtual links established on substrate paths.

We model these networks as undirected connected graphs in which each node and link has a positive real value as the amount of resources. The substrate network is represented by $G_S = (N_S, L_S, R_S^n, R_S^l)$, where N_S , L_S , R_S^n and R_S^l are the set of the substrate nodes, the substrate links, the amounts of node resource, and the amounts of link resource, respectively. Similarly, the virtual network is represented by $G_V = (N_V, L_V, R_V^n, R_V^l)$, but R_V^n and R_V^l are the amounts of resources required.

The VNE problem determines how to map the nodes and links among these networks (Fig. 1). It is defined as a mapping $M : (N_V, L_V) \rightarrow (N'_S, P'_S)$, where N'_S is a subset of N_S , and P'_S is a subset of the set of substrate paths. G_V and (N'_S, P'_S) must have the same topology, and the substrate elements must satisfy the resource requirements of the virtual elements assigned to them. Also, the node mapping must be injective, that is, $\forall n, m \in N_V (n \neq m \Rightarrow M(n) \neq M(m))$.

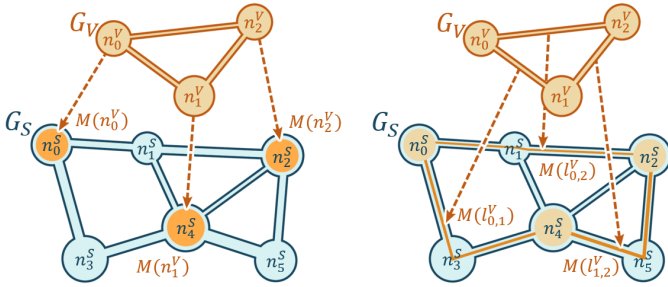


Fig. 1. Model of virtual network embedding

III. METHOD

A. The MAP-Elites algorithm

MAP-Elites [5] is an optimization algorithm that aims to compute the maximum fitness distribution in a low-dimensional feature space for a high-dimensional search space. Users of MAP-Elites need to select several features of interest from the problem object and design a feature space. The feature space has dimensions whose variables are those features and they are divided at a certain granularity, which makes cells in the space; MAP-Elites outputs the solution with the highest fitness in each cell on the feature space. It is reported that MAP-Elites is unlikely to fall into a local optimum compared to conventional genetic algorithms, as it calculates individuals with high fitness in a larger region of the feature space in parallel and generates a new solution based on them.

The pseudocode of MAP-Elites is shown in Algorithm 1. \mathcal{F} and \mathcal{X} are sets, returning fitness and individuals with a cell identifier (ID) as index, respectively. A unique ID for each cell is returned by *feature_descriptor* where the same value is always returned for features contained in the same cell. Each cell contains the individual with the highest fitness

Algorithm 1 Pseudocode of MAP-Elites [5]

procedure MAP-ELITES

$\mathcal{F} \leftarrow \emptyset, \mathcal{X} \leftarrow \emptyset$

for $iter \leftarrow 1, I$ **do**

if $iter < G$ **then**

$x' \leftarrow random_solution()$

else

$x \leftarrow random_selection(\mathcal{X})$

$x' \leftarrow random_variation(x)$

$b' \leftarrow feature_descriptor(x')$

$f' \leftarrow fitness(x')$

if $\mathcal{F}(b') = null$ or $\mathcal{F}(b') < f'$ **then**

$\mathcal{F}(b') \leftarrow f'$

$\mathcal{X}(b') \leftarrow x'$

return *feature-fitness map* (\mathcal{F} and \mathcal{X})

among the individuals explored in it. In each iteration, random variation (mutation and crossover) is applied to a randomly selected individual from the population, and the individual is compared with the current occupant of the cell corresponding to its features, and if it has higher fitness, the occupant is replaced with it, leading to evolution.

B. Encoding of solutions

In dealing with the VNE problem using genetic algorithms, it is necessary to encode the solution of embedding (denoted by M), which is the phenotype, into the genotype. In MAP-Elites, it is known to be effective to directly encode the parameters so that the behavior does not change significantly due to genotypic variation [6], so here we also encode the node-link correspondence directly.

To encode the mappings with a finite fixed gene length, the search space must be limited. In our method, a substrate path assigned to a virtual link is selected from only the k shortest paths. Therefore, a solution mapping can be represented as a combination of $|N_V|$ integers of 1 to $|N_S|$ that indicate substrate nodes mapped to virtual nodes and $|L_V|$ integers of 1 to k that indicate substrate paths mapped to virtual links. Then, the maximum size of the search space is $|N_S|^{|N_V|} \times k^{|N_V|C_2}$, and a mapping in this space can be encoded into an array of integers. In our implementation, we concatenate the binary representation of elements of the array and use the result as a genotype (Fig.2).

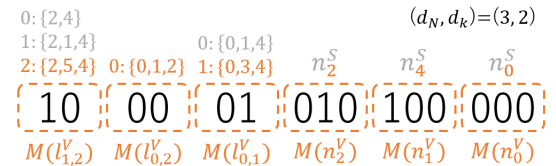


Fig. 2. Genotypes example (encoding the mapping in Fig. 1)

C. Definition of fitness

In genetic algorithms, a population evolves to increase the fitness of individuals to search for a solution, so it is necessary

to define the fitness appropriately to the VNE problem. We define the fitness of the solution M as $F(M)$ in Eq. (1).

$$F(M) = R(M) - \lambda P(M) \quad (1)$$

where $R(M)$ is the total amount of residual resources and $P(M)$ is the penalty term that is the total amount of resource shortage. In our method, there are phenotypes, which are mappings decoded from genotypes, that cannot satisfy the resource requirement, and the penalty term is the summation of the difference between the amount of the substrate resource and the resource requirement in embedding such a phenotype. By setting the value of λ sufficiently large, the population evolves to decrease the penalty term and satisfy resource requirements.

D. Design of the feature space

Since the MAP-Elites algorithm searches for solutions within a feature space of a lower dimension than the problem space, it is important to define a feature space that facilitates efficient solution search. Genetic diversity is known to promote evolution [7], so we should define a feature space such that the coverage in the feature space is positively correlated with the genetic diversity. Additionally, considering the environmental variation in the dynamic VNE problem, it is desirable that diverse feasible solutions are distributed in the feature space so that they can be explored efficiently. Based on these considerations, the following values are used as features in this study.

- The total amount of the physical node resources mapped to the virtual nodes
- The total hops of the physical paths mapped to the virtual links

It is also intended that node and path selections are separated into each feature and that the optimization of each proceeds in parallel, leading to more efficient solution search.

IV. EVALUATION

To examine the influence of solution diversity on adaptability in the dynamic VNE problem, we conducted computer simulations with several genetic algorithms including MAP-Elites to solve the dynamic VNE problem.

A. Simulation Setting

In this simulation, a single virtual network is embedded into a physical network, and a specific physical node becomes unavailable at a certain time. Separate simulations were performed for outages on each of the top 10 physical nodes with the largest resources to evaluate the impact of the amount of resources and the tolerance of each algorithm to it. The duration of the simulation is two epochs, and one epoch is 100 generations. The specific node becomes unavailable at the start of the second epoch.

The physical network have 32 nodes and 60 links, and is generated based on the Barabási-Albert (BA) model [8]. The virtual network is a random graph with 5 nodes and

8 links. For any pair of physical nodes, the 8 shortest paths is calculated using the k shortest path algorithm [9] in advance, and the algorithms determine which of these paths is assigned to the virtual link. Therefore, the size of the search space is $32^5 \times 8^5 C_2 = 2^{55}$.

The resources of the physical network are set to a random number following a log-normal distribution and sorted in descending order of degree centrality. The resources required for the virtual nodes and links also follow the same distribution and multiplied by a constant (0.5 in this paper).

B. Algorithms used for evaluation

To evaluate the effectiveness of the MAP-Elites algorithm, we implemented optimization methods in Table I for comparison. In the following, the notation of these optimization methods follows Table I. The encoding and decoding of the solutions in these genetic algorithms follow the methods defined in Section III-B.

The parameters used in these genetic algorithms are listed in Table II.

TABLE I
OPTIMIZATION METHODS USED IN SIMULATIONS

Notation	Algorithm Description
ME	Our method described in Section III.
EGA	Elitism-based Genetic Algorithm (EGA) is a conventional genetic algorithm [10] that adopts elitism to make its evolution steady. In EGA, the population is replaced with a new population consisting of new individuals by generations. This method maximizes the fitness defined in Section III-C.
SSGA	Steady State Genetic Algorithm (SSGA) is a modified genetic algorithm that replaces the worst individual in the population with a new individual to evolve, not generating a new population. Therefore, there is no generation gap, so the evolution is comparatively steady. This characteristic is in common with the MAP-Elites. This method maximizes the fitness defined in Section III-C.
NSLC	Novelty Search with Local Competition (NSLC) [11] is one of the QD algorithms like MAP-Elites, which is a multi-objective optimization algorithm for novelty and local competitiveness. Novelty is the average Euclidean distance to the k nearest neighbors on the feature space, and local competitiveness is relative performance (fitness defined in Section III-C) in them. This method maximizes the average of novelty and local competitiveness using SSGA.

TABLE II
PARAMETERS OF THE GENETIC ALGORITHMS

Parameters	Value
Gene length	55
Population size (except ME)	100
Feature space granularity (ME)	(10 × 10)
Mutation rate (at each locus)	1%
Crossover rate	90%

C. Simulation results

The simulations were performed 30 times for 100 pairs of different physical and virtual networks for each genetic algorithm, and the results shown below are averages of all simulations.

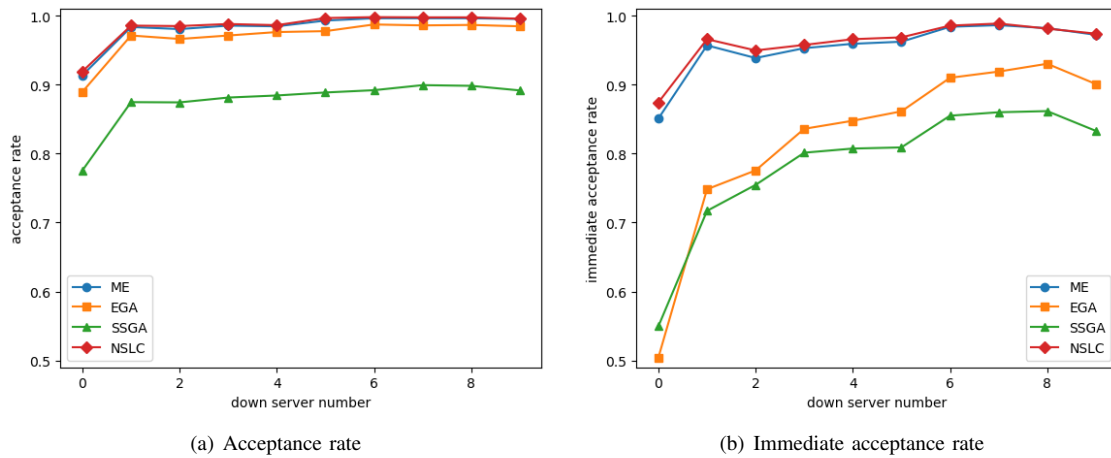


Fig. 3. Solution search performance for each algorithm

The metrics of the performance of the solution search are shown in Fig. 3. The acceptance rate is the percentage of cases where a solution satisfying the constraints is found in the second epoch. The immediate acceptance rate is the percentage of cases in which there is a solution that satisfies the constraints in the second epoch at the time the first epoch ends.

The physical nodes are numbered in descending order of resource amount, and the horizontal axis in Fig. 3 represents the number of the physical node which becomes unavailable in the simulation.

The results show that there is no major difference among the methods in the acceptance rate, but the QD algorithms, namely ME and NSLC, overwhelm the others in the immediate acceptance rate. This means that the QD algorithms search for more diverse solutions in the first epoch, some of which satisfy the constraints in the second epoch, and they lead to a higher tolerance for the environment variation.

Also, the extent of performance degradation of the QD algorithms to the decrease of the horizontal axis is comparatively small, which means that they find the solutions independent of important nodes in more cases.

V. CONCLUSION

Network slicing techniques are becoming more and more important due to the diversification of services running on networks. In this paper, we proposed a dynamic solution method using the QD algorithm for the VNE problem, which is one of the implementation problems. Comparative evaluation through simulations shows that methods using QD algorithms are more effective for the dynamic VNE problem than those using conventional genetic algorithms. We also showed that the ability of QD algorithms to search for diverse and superior solutions leads to their tolerance to unpredictable environmental variations.

However, we did not observe the effectiveness of the search in the feature space designed in this paper. Since the MAP-Elites algorithm was shown to be superior to NSLC-based

methods in its ability to search for high-quality solutions in a wide range of feature space, the design of a feature space that enables a more efficient search in the genotypic domain containing those high-quality solutions is expected to further improve the performance of the method using the MAP-Elites algorithm.

ACKNOWLEDGEMENT

This work was supported by MIC under a grant entitled “R&D of ICT Priority Technology (JPMI00316).”

REFERENCES

- [1] H. Alshaer, “An overview of network virtualization and cloud network as a service,” *International Journal of Network Management*, vol. 25, no. 1, pp. 1–30, 2015.
- [2] I. Jovan, K. Sharif, F. Li, Z. Latif, M. M. Karim, S. Biswas, Y. Wang, and B. Nour, “A survey of network virtualization techniques for internet of things using SDN and NFV,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–40, Apr. 2020.
- [3] H. Cao, S. Wu, Y. Hu, Y. Liu, and L. Yang, “A survey of embedding algorithm for virtual network embedding,” *China Communications*, vol. 16, no. 12, pp. 1–33, Dec. 2019.
- [4] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, pp. 1–17, Jul. 2016.
- [5] J. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *CoRR*, vol. abs/1504.04909, pp. 1–15, Apr. 2015. [Online]. Available: <http://arxiv.org/abs/1504.04909>
- [6] D. Tarapore, J. Clune, A. Cully, and J.-B. Mouret, “How do different encodings influence the performance of the map-elites algorithm?” in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 173–180.
- [7] T. Day, “Information entropy as a measure of genetic diversity and evolvability in colonization,” *Molecular Ecology*, vol. 24, no. 9, pp. 2073–2083, Jan. 2015.
- [8] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [9] J. Y. Yen, “Finding the k shortest loopless paths in a network,” *Management Science*, vol. 17, no. 11, pp. 712–716, Jul. 1971.
- [10] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, pp. 65–85, Oct. 1998.
- [11] J. Lehman and K. O. Stanley, “Evolving a diversity of virtual creatures through novelty search and local competition,” in *Proceedings of annual conference on Genetic and evolutionary computation*, Jul. 2011, pp. 211–218.