Master's Thesis

Title

Dynamic Network Slicing Control Method for Beyond 5G Mobile Network using Quality-Diversity Algorithms

Supervisor

Professor Masayuki Murata

Author

Amato Otsuki

February 2nd, 2024

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Dynamic Network Slicing Control Method for Beyond 5G Mobile Network using Quality-Diversity Algorithms

Amato Otsuki

Abstract

In hosting diverse applications with various service requirements in 5G network, network slicing technology is critical factor, which provides multiple isolated and dedicated virtual network. It is important for the satisfaction of the requests of clients and the profit of providers of the substrate network to allocate the network slices requested by clients efficiently into the substrate network. The optimization problem of substrate resource allocation to network slices is closely related to the virtual network embedding (VNE) problem known as NP-hard problem, thus it is nearly impossible to derive the optimal solution within practical time in a large scale network. In addition, considering a practical use case, it is required to allocate multiple network slices simultaneously, and optimize the allocation of all network slices while new requests arrive sequentially. Therefore, the method which can find a feasible solution immediately in such dynamic environment is essential for network slicing in Beyond 5G network. For the implementation of the mechanism of deriving a new solution adaptively for dynamic environment utilizing the previous solutions, genetic algorithm which is an optimization algorithm based on a model of organism evolution can be applied. As genetic algorithm can deal with general combinatorial optimization problems, we modeled the problem of deciding the mapping between the nodes or links of the substrate network and those of network slices as a combinatorial optimization problem, and realize dynamic network slice embedding applying genetic algorithm to this problem and finding feasible solutions. However, it is known that conventional genetic algorithm tends to lose solution diversity, and Quality-Diversity (QD) algorithms are proposed as methods overcoming this problem, which output solutions with diverse characteristics. It seems to be desirable to maintain solution diversity for the

immediate adaptation to dynamic environment, so we proposed a dynamic network slice control method using QD algorithm. When multiple network slices are controlled integratedly, allocation of elements of all network slices must be considered simultaneously, and the scale of problem space increase exponentially for the number of network slices. Thus, for the scalability, it is necessary to assign a separate controller to each network slice and control them separately. On the other hand, it is unable to achieve total optimization only optimizing each slice separately because it may lead to the degradation of quality of service (QoS) of other slices in B5G network there are the diverse service requirements. Hence, introducing unselfish evaluation values additionally as a multi-objective optimization, we attempt to realize the cooperative control considering other slices while controlling slices separately. We model temporal arrival of network slice requests as Poisson process and evaluated the proposed methods through computer simulation. We assess the degree of total optimization through the acceptance rate of network slice requests and compared our methods with methods using conventional genetic algorithm and heuristic proposed in previous work. As a result, it is shown the acceptance rate of our methods was higher than other methods. Moreover, we reveal the effects of unselfish evaluation values through simulation under static environment.

Keywords

Beyond 5G network

Network slicing

SDN (Software defined network)

VNE (Virtual network embedding)

Genetic algorithm

Quality-Diversity algorithm

MAP-Elites

Contents

List of Figures

List of Tables

1 Introduction

There are various use cases in the 5G network, and the network requirements vary by the applications running on them. As new network services appear daily and their usage conditions change, it is not realistic to build a dedicated network for each service. Therefore, one of the most important factors is network slicing technology, which virtually divides physical resources on a shared substrate network and constructs multiple virtual networks on the substrate network. As multiple clients request network slices with various requirements, network providers are required to deploy those slices within finite physical resources. It is important for both clients and providers to efficiently deploy more slices under the resource requirements.

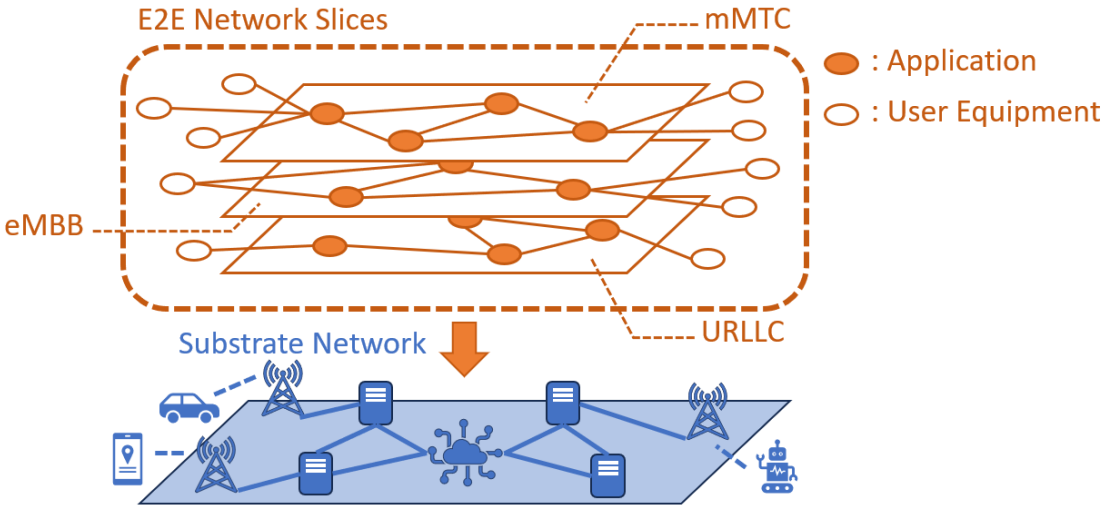

Figure 1: Diagram of E2E network slicing in 5G network

Although there have been studies on network slicing in 5G networks from various perspectives, few have treated it as a resource allocation problem assuming end to end (E2E) communication between user equipment(UE) and applications deployed on cloud servers [1]. When network slicing is considered as a resource allocation problem, it is required to optimize the allocation of the substrate resources in terms of acceptance ratio and cost while satisfying the requirements of each slice. This problem is closely related to the virtual network embedding (VNE) problem, which has been studied extensively [2], and formulated in [3] as network slice embedding(NSE) problem based on the VNE problem. The charac-

teristics of 5G network slicing assuming E2E communication are that the requirements for slices vary significantly depending on the service, and that it targets the construction of networks between UE and cloud servers. Compared to embedding multiple homogeneous virtual networks, when the requirements of each slice are significantly different, only selfish selection is likely to degrade the QoS of the other slices and fail to meet service level agreement (SLA) and unable to achieve total optimization.

NSE problem is a combinatorial optimization problem, and methods for solving it using linear programming solver and heuristics based on node ranking have been proposed [2]. Linear programming solver is not scalable to the size of network, and computation time is a trouble in practical use. In addition, in practical use cases, multiple slices need to be optimized in parallel as new slice requests arrive over time. In order to find solutions adaptively to such a dynamic environment in a short time, a method is required that can find the optimal solution for the new environment by utilizing previous solutions.

The mechanism to adapt to a certain environment by modifying the solution at that time can be regarded as a model of organism evolution, and one of the optimization algorithms based on the model is genetic algorithm. Genetic algorithm is one of the metaheuristics that can handle general combinatorial optimization problems and has been used in a wide range of problems because it can handle optimization problems independently of the properties of the objective function. The previous research [4], which applies the genetic algorithm to the VNE problem, shows the acceptance ratio of the method is higher than that of the typical VNE method based on mixed integer programming [5, 6]. Since the genetic algorithm allows the objective function to be arbitrarily designed, it is useful in NSE problems to satisfy various requirements simultaneously and to achieve the goal of total optimization.

On the other hand, it is known that the population converges to a single form in conventional genetic algorithms, unlike in nature where diverse forms can be found. For rapid adaptation to environmental variations in the NSE problem, it is required to maintain diverse solutions in the population. In response to this, methods have been proposed that can generate diverse and high-quality solutions by proceeding with optimization while maintaining solution diversity, and such methods are called quality-diversity algorithms. MAP-Elites (Multi-dimensional Archive of Phenotypic Elites) [7] and NSLC (Novelty

Search with Local Competition) are representative examples, and our group's paper applying them to the dynamic VNE shows [8]. Because of its lower computational cost, in this paper, we applies the MAP-Elites algorithm to the NSE problem and proposes a method of optimizing multiple network slices in parallel.

The genetic algorithm treats the solution as if it were an individual organism, and proceeds with optimization by repeatedly generating and selecting new individuals from a randomly generated population. Each individual has a sequence of integers called the genotype, and the genotype must be decodable into a phenotype, or solution, by some procedure. In addition, each individual has an evaluation value that determines whether or not it can survive in its environment, which is called its fitness. The genotypes can be changed by genetic manipulation, such as crossover with the genotype of another individual or mutation, to produce new individuals with new genotypes. Their fitness is calculated from the solutions obtained by decoding their genotypes, and the values determines whether they remain in the population.

In order to apply the MAP-Elites algorithm to the NSE problem, it is necessary to define the encoding method of the solution and the fitness. In MAP-Elites, it is known the encoding method that directly encodes parameters as genotypes is effective [9], and in the NSE problem, the correspondence of the substrate node/path to each virtual node/link in the network slice can be used as the genotype. Since it is important to embed the network slice in the NSE problem by satisfying the requirements and saving the substrate resources, the fitness is defined as a value that decreases when the slice requirements are violated and increases when the residual resources are large.

As mentioned above, in a realistic scenario of the NSE problem, it is required to be embed multiple network slices in parallel. If embedding of all network slices is controlled by a single controller, the size of the problem space increases exponentially with respect to the number of network slices, since combinations about elements of all network slices must be handled. Therefore, it is not realistic to control multiple network slices with a single controller from the perspective of scalability. Therefore, in this method, a controller using the MAP-Elites algorithm is assigned to each network slice separately, and optimization of each network slice is performed in parallel. However, it is unable to achieve total optimization by simply optimizing each individual slice. Therefore, by introducing unselfish

evaluation values into the fitness, we aim to realize cooperative behavior and achieve total optimization.

We evaluated the effectiveness of this method through computer simulation. We modeled the arrival of network slice requests as a Poisson process, quantified the degree of total optimization through the acceptance ratio, and compared the methods introducing unselfish evaluation values and not in a simulation program in which multiple network slices generated over time are controlled by individual controllers in parallel.

The remainder of this paper is organized as follows. Section 2 introduces the overview of NSE problem and QD algorithms, and the detail of MAP-Elites, the QD algorithm used in this research. Section 3 formulates the NSE problem in detail and describes the models used in the simulation. Section 4 explains and formulates our method. Section 5 shows the simulation settings and the results of computer simulations. Section 6 summarizes this paper.

2 Related work

2.1 Network Slice Embedding Problem

As the use cases in 5G networks have diversified in recent years, so have the requirements for the networks. Specifically, there are such use cases as URLLC (Ultra-reliable and Low Latency Communications), which requires high reliability and low latency for applications such as automated driving and remote medical robots; eMBB (enhanced Mobile Broadband), which requires high data rates for such applications as video streaming; and mMTC (Massive Machine Technology), which requires simultaneous communications in large-scale networks such as sensor networks and IoT [10]. Cooperative control is essential to efficiently deploy slices with different requirements simultaneously. On the other hand, integrated control of all slices for total optimization leads to an increase in the problem scale and makes it difficult to solve the problem in a practical time. Therefore, while each slice is controlled individually, it is required to cooperate with each other to improve the overall acceptance ratio of network slice requests.

When the control of network slicing in 5G networks is regarded as an optimization problem, it has various algorithmic aspects, one of which is as a resource allocation problem [1]. In [11], network slice optimization is defined as a Virtual Network Embedding (VNE) problem, but the target of control is from the base station to the core cloud, not E2E communication including UE. In addition, Chien et al. [12] assumes E2E communication on 5G networks and implements resource allocation control for network slices, but the slice embedding is not subject to control, and the resource allocation algorithm does not consider optimization. Ludwig et al. [2] defines the optimization of network slice embedding for E2E communication including UE as the NSE problem, which is an extension of the VNE problem.

In the NSE problem, the substrate network and network slices are modeled as undirected graphs, and the nodes composing the substrate network are classified as UE and cloud nodes, while the nodes comprosing the network slice are classified as UE and applications. Cloud nodes and substrate links have finite resources, which are consumed to host applications and communications between UE and applications. The NSE problem is to determine which requests to accept from multiple network slices and the way to allocate

resources to those slices under this resource constraint. In [3], the NSE problem including application chains is formulated, and the definition in this paper is mainly based on the description in this work. However, we exclude the terms of availability and reliability that are defined in this work.

In [2], an extension of the NodeRank method is proposed, which is a heuristics for the VNE problem, and shows high scalability in terms of computation time, while having similar acceptance ratio of slice requests compared to an existing method based on integer linear programming. On the other hand, even though this work includes latency in the slice requirements, the proposed method does not consider latency. In addition, although the proposed method assumes the parallel embedding of multiple network slices, it does not incorporate the network resource condition varying over time and information from other slices into the ranking, and does not consider the total optimization.

The aim of this research is to realize a parallel optimization method that controls network slices individually to ensure scalability, while cooperating with each other to achieve total optimization. Taking advantage of the flexibility in designing the fitness (objective function) and search space of the quality diversity algorithm, we incorporate the terms about latency requirements and necessary for cooperation with the controller managing other slices into the fitness and design a search space effective for improving the fitness to achieve efficient total optimization.

2.2 Quality-Diversity Algorithm

Conventional optimization algorithms search for a single global optimal solution that maximizes the objective function, and multi-objective optimization algorithms also search for vertices in the search space. Quality diversity algorithm is evolutionary algorithmic framework that differ from them in that it exhaustively searches for qualitatively different and superior solutions distributed throughout the search space [13]. The main difference from multi-objective optimization algorithms is that they search within the feature space rather than the genotype space, and they also search in regions of the feature space where there are no vertices of the objective function. This concept originated in the novelty search [14], and algorithms that divergently search for superior solutions in the search space, such as NSLC [15] and MAP-Elites [7], were developed, and the quality diversity algorithm has

been proposed as the framework of such algorithms [16]

In optimization with the QD algorithms, it is necessary and important to define the fitness which is the evaluation value of the solution and the feature space. Since the population evolves to improve the fitness, it is necessary to define the fitness appropriately in order to achieve the objective. It is also important to design the feature space so that various solutions are widely distributed, since solutions are divergently searched in the feature space.

2.2.1 MAP-Elites Algorithm

MAP-Elites [7] is a sort of QD 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.

MAP-Elites is inspired by novelty-search [15] with local competition: since MAP-Elites seeks only a single optimal solution for each cell (this solution is called an occupant), if the newly generated solution belongs to a cell that already has an occupant, the solution with the higher fitness of them remains as the occupant. This property reflects the fact that selection occurs through local competition with neighboring individuals in nature.

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 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.

Algorithm 1 Pseudocode of MAP-Elites [7]

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

$x \leftarrow random\_selection(\mathcal{X})$

$x' \leftarrow random\_variation(x)$

$\boldsymbol{b}' \leftarrow feature\_descriptor(x')$

$f' \leftarrow fitness(x')$

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

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

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

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

3 Formulation of Dynamic Network Slice Embedding Problem

In this research, the NSE problem is formulated mainly based on the previous work [2] [3]. The substrate network and network slices are modeled as undirected graphs. A undirected graph is expressed generally as a ordered pair $G = (V, E)$, where V is the set of vertices, and E is the set of edges. The edge between two vertices $v_i, v_j \in V$ is denoted by $e_{i,j} = (v_i, v_j)$. Then, the substrate network is represented by $N_S = (V_S, E_S)$. The vertices of substrate network is classified into UE and cloud nodes, they are denoted by V_U, V_C, respectively, and $V_S = V_U \cap V_C$. E_S represents the wired or wireless links, and there is no connection between UE, which means $\forall e_{i,j} \in E_S \, (v_i \in V_C \vee v_j \in V_C)$. The cloud nodes and the wired/wireless links have finite resources. CPU resource $C(v)$ and memory capacity $M(v)$ of a cloud node $v \in V_C$, and throughput $T(e)$ and latency $L(e)$ of a link $e \in E_S$ are considered in this research. In addition, all cloud nodes have a loop with infinite throuput and zero latency. This is because the situation is assumed that each cloud node consists of multiple nodes and each of them can host application individually.

Similarly, the requested network slice is represented by $N_R = (V_R, E_R)$. The network slice consists of some of the UE in the substate network and applications hosted by cloud nodes. Thus, let $V'_U \subset V_U$ and V_A denote the set of the UE used in this slice and the applications, respectively, and $V_R = V'_U \cup V_A$. E_R is the set of virtual links between UE and applications, and the is no virtual link between UE, which means $\forall e_{i,j} \in E_R \, (v_i \in V_A \vee v_j \in V_A)$. The applications and the virtual links have attributes of the resource requirements corresponding to the substrate resources(C, M, T, L).

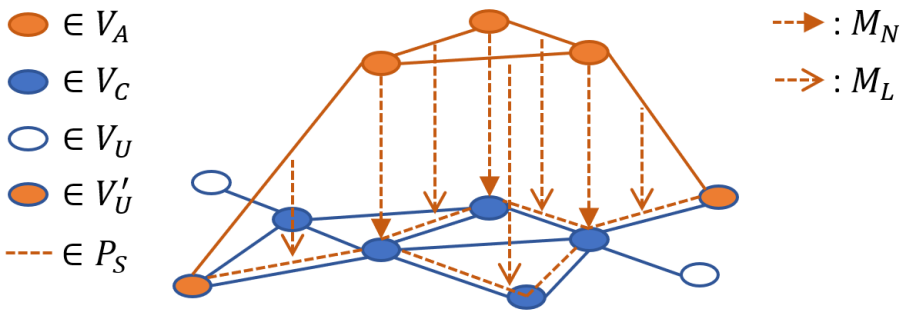

Figure 2: Diagram of NSE problem

In the NSE problem, for multiple network slice requests, it is required to decide whether each of them is accepted or not and how to allocate resources. Let $\mathcal{R}$ denote the set of the N requested network slices, and for each element $N_R^k = (V_R^k, E_R^k) = (V_U'^k \cup V_A^k, E_R^k)(1 \leq k \leq N)$, whether it is accepted or not is represented by the mapping $M_A \colon \mathcal{R} \to \{0, 1\}$. In addition, the set of the numbers of accepted network slices is denoted by $\mathbb{N}_A = \{k \mid M_A(N_R^k) = 1\}$.

$$M_A(N_R^k) := \begin{cases} 1 & \text{if } N_R^k \text{ is accepted} \\ 0 & \text{otherwise} \end{cases}$$

For each element N_R^k of the set of accepted network slices $\mathcal{R}_A = \{N_R^k \in \mathcal{R} \mid M_A(N_R^k) = 1\}$, the way to allocate its applications to cloud nodes is represented the mapping $M_N^k \colon V_A^k \to V_C$. To extend the domain to V_R^k, the mapping $M_N'^k \colon V_R^k \to V_S$ is defined additionally. It should be noted that the mapping M_N^k does not have to be injective while the node allocation in VNE problem does so.

$$M_N'^k(n) := \begin{cases} M_N^k(n) & (n \in V_A^k) \\ n & (n \in V_U'^k) \end{cases}$$

The way to allocate virtual links to substrate paths is represented by the mapping $M_L^k \colon E_R^k \to P_S$, where P_S is the set of all possible path on the substrate network. A path is a finite sequence of vertices, the kth term(vertex) of a path p is denoted by p_k, and its length is denoted by $|p|$. In addition, the notation $e_{i,j} \overset{\text{in}}{\sim} p$ means that the edge $e_{i,j}$ is included in the path p, and the latency of the whole path p is denoted by $L_P(p)$, which is formulated by

$$e_{i,j} \overset{\text{in}}{\sim} p \overset{\text{def}}{\Leftrightarrow} \exists q \left((p_q = v_i \wedge p_{q+1} = v_j) \vee (p_q = v_j \wedge p_{q+1} = v_i) \right)$$

$$L_P(p) := \sum_{k=1}^{|p|-1} L\left((p_k, p_{k+1})\right)$$

The allocated network slice must have the same topology as the requested network slice, and this constraint is defined as the logical formula 1.

$$\forall k \in \mathbb{N}_A \; \forall e_{i,j} \in E_R^k \left(M_L^k(e_{i,j})_1 = M_N'^k(v_i) \wedge M_L^k(e_{i,j})_{|M_L^k(e_{i,j})|} = M_N'^k(v_j) \right) \tag{1}$$

Moreover, each cloud node must satisfy the resource requirements of applications it hosts, and this constraint is defined as the logical formula 2.

$$\forall c \in V_C \; \forall Attr \in \{C, M\} \left(\sum_{k \in \mathbb{N}_A} \sum_{a \in V_A^k(c)} Attr(a) < Attr(c) \right) \tag{2}$$

where

$$V_A^k(c) = \{a \in V_A^k \mid M_N^k(a) = c\}$$

Similarly, each substrate link must satisfy the resource requirements of the virtual links established on it, and this constraint is defined as the logical formulae 3,4.

$$\forall e_{i,j} \in E_S \left(\sum_{k \in \mathbb{N}_A} \sum_{e_r \in E_R^k(e_{i,j})} T(e_r) < T(e_{i,j}) \right) \tag{3}$$

$$\forall k \in \mathbb{N}_A \; \forall e_r \in E_R^k \left(L_P(M_L^k(e_r)) < L(e_r) \right) \tag{4}$$

where

$$E_R^k(e_{i,j}) = \{e_r \in E_R^k \mid e_{i,j} \overset{\text{in}}{\sim} M_L^k(e_r)\}$$

The NSE problem is defined as deciding the mappings M_A, M_N^k, M_L^k satisfying these logical formula.

In the following subsections, the models about the NSE problem is explained assumed in the evaluation of this research.

3.1 Substrate Network Model

In this research, the substrate network architecture is assumed to be the layer substrate consisting of four layers: UE, Node B, edge cloud, and central cloud, proposed in [2], [17]. The layers except for UE are classified as cloud nodes, the resource amount increases, and the number of nodes decrease in the above order. There are only links connecting between adjacent layers. Let V_{NB}, V_{EC}, V_{CC} denote the sets of nodes of Node B, edge cloud, central cloud, respectively, this is formulated by

$$\forall e_{i,j} \in E_S \left((i \in V_U \wedge j \in V_{NB}) \vee (i \in V_{NB} \wedge j \in V_{EC}) \vee (i \in V_{EC} \wedge j \in V_{CC}) \right)$$

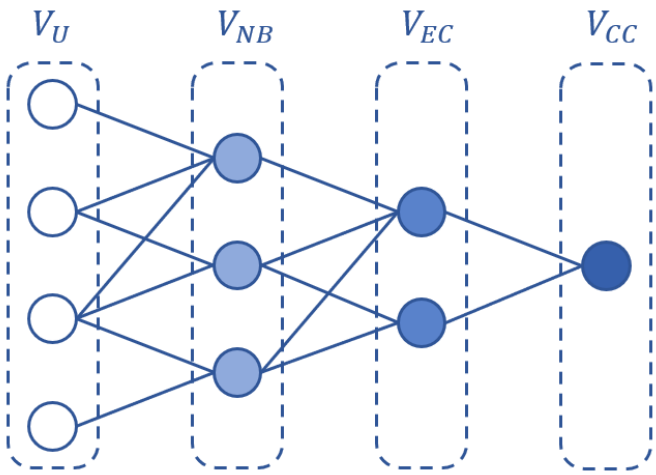

Figure 3: Layer substrate network model

Although the wireless links between UE and Node B are modeled similarly to other wired links in [2], the available bandwidth is shared among UE connecting to the same Node B in actual, thus they should be modeled such that the throughput is shared. In this paper, we added the layer representing radio access network(RAN) between UE layer and Node B layer, and UE connecting to the same Node B is made through the same RAN to share the available throughput. Here, $|V_{NB}| = |V_{RAN}| \wedge \forall v_i \in V_{NB}\ \exists! v_j \in V_{RAN}(e_{i,j} \in E_S)$. In addition, the links between Node B and RAN have certain throughput and zero latency, and the links between UE and RAN have infinite throughput and certain latency.

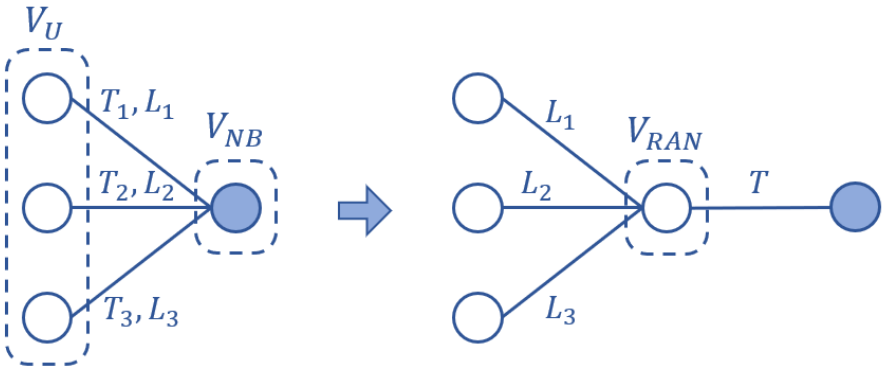

Figure 4: Addition of RAN layer

3.2 Network Slice Request Arrival Model

Network slice requests is supposed to be issued individually by each client, the issues are events independent of each other, and such process can be modeled as Poisson process. The occurrence probability of events in Poisson process follows Poisson distribution, which is a discrete probability distribution that express the probability of the case the event, which occurs λ times per time unit on average, occurs k times within a time unit. Let X be the random variable representing the number of events that occurs within a time unit, the probability is

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Network slices are deactivated after a certain interval from the time they are hosted, and the interval is called lifetime. The interval t between occurrence of events in Poisson process follows the exponential distribution with the below probability density function.

$$f(t) = \lambda e^{-\lambda t}$$

Therefore, the model in which n network slice requests is alive simultaneously can be implemented by setting the lifetime of slices as the random number following the same distribution as the interval of event which occurs λ/n times per time unit on average. Thus, the lifetime of each network slice is generated as random number following the exponential distribution with the below probability density function.

$$f(t) = \frac{\lambda}{n} e^{-\frac{\lambda}{n} t}$$

4 Dynamic Network Slice Control Method using QD algorithm

In the following explanation, the formulae defined in section 3 are used. In this method, the behavior of the controller of embedding individual network slice is described with MAP-Elites algorithm, and the controllers assigned to network slices optimize their own embedding separately in parallel. In short, the controllers optimize the mappings M_A, M_N^k, M_L^k that is the solution of the NSE problem.

The controller of the network slice N_R^k is defined as $\mathcal{C}^k$.

$$\mathcal{C}^k = (\mathcal{P}^k, F^k, A^k)$$

where $\mathcal{P}^k$ is the population in MAP-Elites algorithm, which is the set of the genotype of an individual in each cell(occupants) of the feature space, and F^k, A^k are the mappings from the genotype to the fitness and the value representing whether it is feasible solution or not, respectively. A feasible solution mentioned here is the solution satisfying the logical formulae 1–4, and $A^k(g) = 1$ if the genotype g is a feasible solution, otherwise $A^k(g) = 0$. The details of the mappings F^k, A^k are described in section 4.2. $\mathcal{P}^k$ varies every iteration in Algorithm 1, so the state of the controller at the time t is denoted by $\mathcal{C}_t^k = (\mathcal{P}_t^k, F^k, A^k)$.

Here, the mapping M_A at the time t is defined as

$$M_A(N_R^k \mid t) := \begin{cases} 1 & \text{if } \exists i \leq t \ \exists g \in \mathcal{P}_i^k (A^k(g, i) = 1) \\ 0 & \text{otherwise} \end{cases}$$

which means the network slice N_R^k is accepted if the controller $\mathcal{C}^k$ has found a feasible solution by the time t.

The genotypes which are elements in the population $\mathcal{P}_t^k$ can be translated into the pair of mappings corresponding to M_N^k, M_L^k by certain procedure, and the details is mentioned in section 4.1. Let us define this procedure as the mapping dec^k here, and the mappings

M_N^k, M_L^k at the time t are decided as

$$M_N^k(a \mid t) := M_{Nt}^k(a)$$
$$M_L^k(l \mid t) := M_{Lt}^k(l)$$
$$(M_{Nt}^k, M_{Lt}^k) = dec^k\left(\underset{g \in \mathcal{A}_{t'}^k}{\operatorname{argmax}} F^k(g, t')\right)$$

where

$$t' = \max\{i \leq t \mid \exists g \in \mathcal{P}_i^k(A^k(g, i) = 1)\},\ \mathcal{A}_{t'}^k = \{g \in \mathcal{P}_{t'}^k \mid A^k(g, t') = 1\}$$

These formulae means the solution at the time t is made by decoding the genotype of the individual with the highest fitness of the feasible solutions found last by the time t.

4.1 Encoding of Solutions

In dealing with combinatorial optimization problem using genetic algorithms, the genotype and the way to decode it into solution as phenotype must be defined. It is known that it is effective to encode the parameters directly into genotype with MAP-Elites, so the solution parameters of the NSE problem are used as the genotype in this research. The solution in optimization of the network slice N_R^k is the mappings M_N^k, M_L^k, and the parameters are the substrate elements mapped from $a \in V_A^k$ and $l \in E_R^k$. Therefore, the genotype g is defined as the sequence of the values representing these elements. For nodes, when $M_N^k(a_i) = c_j(a_i \in V_A^k, c_j \in V_C^k)$, g_i is set to j. For edges, it is supposed that the substrate path assigned to each virtual link is selected from only the K shortest paths between the substrate nodes assigned to the ends of the link, and the number of path when arranged in the ascending order of length is used as parameter. Let $p_{i,j}^h$ denote the h-th shortest path between c_i, c_j, and it is supposed that each element of E_R^k can be denoted by l_d using the unique number $d \in \mathbb{N}_{\leqslant |E_R^k|}$. Finally, when $M_L^k(l_d) = p^h_{M_N^k(a_i), M_N^k(a_j)}(l_d = (i, j) \in E_R^k)$, $g_{|V_A|+d}$ is set to h. Thus, any genotype is an element of the set $\mathcal{G}^k = \mathbb{N}_{\leqslant |V_C|}^{|V_A^k|} \times \mathbb{N}_{\leqslant K}^{|E_R^k|}$, and the search space of the algorithm is the set $\mathcal{G}^k$.

Conversely, when the genotype $g \in \mathcal{G}^k$ is decoded into the pair of mappings corresponding to M_N^k, M_L^k, each term of the genotype g is translated to the substrate elements

mapped by the mappings. Thus, the mapping dec^k is defined as

$$dec^k(g) := (\phi_g^k, \psi_g^k)$$
$$\phi_g^k(a_i) := c_{g_i}$$
$$\psi_g^k(l_d) = \psi_g^k((a_i, a_j)) := p^{g_{|V_A^k|+d}}_{\phi_g^k(a_i),\phi_g^k(a_j)}$$

Note that the pair of mappings ϕ_g^k, ψ_g^k decoded by dec^k does not satisfy the logical formulae 1–4 necessarily, and phenotypes not satisfying these constraints are not feasible solutions.

4.2 Definition of Fitness

The fitness which is the objective function should be defined appropriately in order to reflect the requirements. In this method, which optimizes multiple network slices in parallel, when an individual is evaluated in the controller $\mathcal{C}^k$, the slices of the other controllers should be embedded according to the solution made by them at that time. At the time t, the residual resource amounts R_N of $c \in V_C$ and the residual throughput of $e \in E_S$ are defined as

$$R_N(c, Attr, k, t) := Attr(c) - \sum_{n \in \mathbb{N}_A^k} \sum_{a \in V_A^n(c|t)} Attr(a)$$
$$R_L(e, k, t) := T(e) - \sum_{n \in \mathbb{N}_A^k} \sum_{e_r \in E_R^n(e|t)} T(e_r)$$

where

$$Attr \in \{C, M\},\ \mathbb{N}_A^k = \{n \mid n \neq k \wedge M_A(N_R^n \mid t) = 1\},$$
$$V_A^n(c \mid t) = \{a \in V_A^n \mid M_N^n(a \mid t) = c\},\ E_R^n(e \mid t) = \{e_r \in E_R^n \mid e \overset{\text{in}}{\sim} M_L^n(e_r \mid t)\}$$

Also, the requested resource amounts Q_N to $c \in V_C$ and the requested throughput Q_L to $e \in E_S$ when the network slice N_R^k is embedded according to the pair of mappings $dec^k(g)$ are defined as

$$Q_N(c, Attr, k, g) := \sum_{a \in V_A^k(c|g)} Attr(a)$$
$$Q_L(e, k, g) := \sum_{e_r \in E_R^k(e|g)} T(e_r)$$

where

$$V_A^k(c \mid g) = \{a \in V_A^k \mid \phi_g^k(a) = c\},\ E_R^k(e \mid g) = \{e_r \in E_R^k \mid e \overset{\text{in}}{\sim} \psi_g^k(e_r)\}$$

The essential requirement in NSE problem is satisfaction of the logical formulae 1–4, so the fitness should be defined such that it decrease as these constraints are violated, and then the population evolves to satisfy the requirement. The logical formula 1 is always satisfied essentially by the definition of genotype mentioned in section 4.1, the logical formulae 2–4 only have to be taken into account. Therefore, the penalty term P^k representing the degree of the violation of the resource requirements is defined as below. In this term, the term about latency constraint should be weighted significantly since the formula 4 is independent of the other slice embedding, and the satisfaction of this is top priority. Thus, setting the constant μ large directs the evolution to satisfy the latency constraint preferentially.

$$\begin{aligned} P^k(g \mid t) := & \sum_{X \in \{C,M\}} \sum_{c \in V_C} \max\{Q_N(c, X, k, g) - R_N(c, X, k, t), 0\} \\ & + \sum_{e \in E_S} \max\{Q_L(e, k, g) - R_L(e, k, t), 0\} \\ & + \mu \sum_{e_r \in E_R^k} \max\left\{L_P(\psi_g^k(e_r)) - L(e_r), 0\right\} \end{aligned}$$

Here, when $P^k(g \mid t) = 0$, the pair of mappings $dec^k(g)$ is a feasible solution of the NSE problem. Therefore, the mapping A^k is defined as

$$A^k(g, t) := \begin{cases} 1 & \text{if } P^k(g \mid t) = 0 \\ 0 & \text{otherwise} \end{cases}$$

Additionally, there is a basic objective in NSE problem is to save the cost of hosting network slices. This can be achieved by minimizing the total requested resource amount. Since the amounts of requested resources of cloud nodes from applications are constant, it is needed to consider only the requested throughput from virtual links.

Therefore, the fitness reflecting these objective can be composed as below.

$$-\rho P^k(g \mid t) - \sum_{e \in E_S} Q_L(e, k, g)$$

where ρ is a constant, and setting this large leads to the evolution such that achieve the resource requirements preferentially.

However, this definition of fitness results in just local optimization about the network slice N_R^k since the other slices is not taken into account. To overcome this problem and realize the cooperative behavior considering the other slices, we defined the unselfish evaluation values additionally and incorporate them into the fitness.

4.2.1 Unselfish Behavior: Load Balancing

Generally, to accept more network slices, it is important to avoid the resource fragmentation, that is, to reduce the number of bottleneck, which is element with little residual resource. This can be realized by using the elements with abundant residual resources preferentially. The total residual resources of the substrate elements assigned to the network slice N_R^k is incorporated into the fitness, thus such solutions are selected preferentially. Therefore, the mapping B^k is defined as

$$B^k(g \mid t) := \sum_{X \in \{C,M\}} \sum_{a \in V_A^k} R_N(\phi_g^k(a), X, k, t) + \sum_{e_r \in E_R^k} \sum_{e \in E_S(\psi_g^k(e_r))} R_L(e, k, t)$$

where

$$E_S(p) = \{e \in E_S \mid e \overset{\text{in}}{\sim} p\}$$

4.2.2 Unselfish Behavior: Giving Way

When $M_A(N_R^k \mid t) = 1$, if the controller $\mathcal{C}^k$ searches for such feasible solutions that do not use the substrate element where the demand from not accepted slices concentrates, it is expected more slices is accepted. The individual with the highest fitness in each controller is the allocation best adapting to the current resource condition and highly possible to be accepted later. Therefore, the total requested resource amounts by these individuals are regarded as the demand of the substrate elements. At the time t, the values representing

the demand of $c \in V_C$ and $e \in E_S$ denoted by D_N, D_L, respectively, are defined as

$$D_N(c,k,t) := \sum_{X \in \{C,M\}} \sum_{n \in \mathbb{N}_R^k} \sum_{a \in V_A^n(c|t)} Attr(a)$$
$$D_L(e,k,t) := T(e) - \sum_{n \in \mathbb{N}_R^k} \sum_{e_r \in E_R^n(e|t)} T(e_r)$$

where

$$\mathbb{N}_R = \{n \mid M_A(N_R^n \mid t) = 0\},\ V_A^n(c \mid t) = \{a \in V_A^n \mid \phi_{g_t^n}^n(a) = c\},$$
$$E_R^n(e \mid t) = \{e_r \in E_R^n \mid e \overset{\text{in}}{\sim} \psi_{g_t^n}^n(e_r)\},\ g_t^n = \underset{g \in \mathcal{P}_t^n}{\operatorname{argmax}} F^k(g,t)$$

The controller $\mathcal{C}^k$ should search for feasible solutions using the substrate elements where these values are lower, this means the population should evolve to minimize the total demand of the substrate elements assigned to the slice. Thus, the mapping D^k is defined as

$$D^k(g \mid t) := \sum_{a \in V_A^k} D_N(\phi_g^k(a), k, t) + \sum_{e_r \in E_R^k} \sum_{e \in E_S(\psi_g^k(e_r))} D_L(e,k,t)$$

where

$$E_S(p) = \{e \in E_S \mid e \overset{\text{in}}{\sim} p\}$$

4.2.3 Definition of Fitness for Cooperative Control

Using the terms defined above, the mapping returning the fitness F^k is defined as

$$F^k(g,t) := -\sum_{e \in E_S} Q_L(e,k,g) - \rho P^k(g \mid t) + \epsilon B^k(g \mid t) - M_A(N_R^k \mid t) \cdot \delta D^k(g \mid t)$$

where ϵ, δ are constants, ϵ should be set small for the third term not to affect the other requirements. The fourth term is valid only when a feasible solution has been already found in the controller $\mathcal{C}^k$, and set large to give way to the other controllers.

4.3 Design of Feature Space

Since MAP-Elites algorithm searches for solutions in feature space, it is important to design such feature space that can capture the superior solutions spread in search space

well. In addition, for the flexible adaptation to the environmental change occurring in dynamic NSE problem, it is effective to use the features whose the optimal value varies by the environment as variables. Based on above, the features below are used in this method.

1. The total resource amount of assigned nodes

    — This feature is the total substrate resource amount of the cloud nodes assigned to the applications. In the layer substrate network model defined in section 3.1, the resource amounts varies by the type of cloud node. Therefore, this feature reflects which type of nodes are mainly used. Since the optimal node usage varies by the service requirements of the slice and the substrate resource condition, it is effective to search for solutions divergently about this feature. However, the distribution of the solutions in the dimension whose variable is this value is heterogeneous, which leads to the degradation of solution search efficiency. Thus, the value F_1 is used as the substitution of this value. Note that the numbers of the substrate nodes are assigned in order of Node B, edge cloud, and central cloud.

2. The total latency

    — This feature is the total latency of the substrate paths assigned to the virtual links. This value is correlated mainly with the number of hops of paths, but the paths with the same number of hops can be distinguished by latency, which leads to the diverse path selection. Since the diverse path selection results in avoiding usage of resources with high demand, it is effective in this problem. For this feature, the value F_2 is used as the substitution for the same reason as above.

$$F_1(g,k) := \sum_{i=1}^{|V_A^k|} g_i$$

$$F_2(g,k) := \sum_{i=1}^{|E_R^k|} g_{|V_A^k|+i}$$

5 Evaluation

For the dynamic NSE problem of controlling multiple network slices in parallel, we compared our proposed method with the previous methods through computer simulations in the two network slice request scenarios below.

Scenario 1

The scenario based on the network slice arrival model defined in 3.2. The parameters of the model are set as $\lambda = 0.2, n = 40$, and the duration of a single simulation is 1000 time units. The time-out period is set to 100 time units. If a controller does not find a feasible solution by the period, the network slice request is rejected. This scenario is used as dynamic environment assuming realistic use case.

Scenario 2

The scenario where 100 network slice requests arrive at the beginning of simulation. This is used for the evaluation of solution search efficiency and total optimization performance of each method under static environment.

In any simulation, 100 individuals (solution candidates) are evaluated per time unit. In this paper, the real-time evaluation is not performed since the simulation program is not optimized. It is assumed there is no difference about computation time among the methods because the evaluation process is same in all methods.

The methods we evaluated are shown in Table 1. The proposed methods are notated with * in this table. The parameters in Table 1 are constants in the equation 4.2.3. Elitism-based Genetic Algorithm (EGA) is a basic genetic algorithm that adopt elitism to make its evolution steady and used as a conventional genetic algorithm in this comparison. The method NR applies RW-BFS which is a heuristic in the VNE problem to this problem in the same way as the heuristic proposed in [2].

The substrate network is generated based on the layer substrate network model defined in section 3.1 and common between the scenarios. The parameter settings of the substrate network used in simulation are shown in Table 2. In Table 2, $E_{A,B}$ is the set of links between elements of V_A, V_B, and $|E|$ is the number of elements of V_B connected to each element of V_A. Note that elements of $E_{U,NB}$ are through V_{RAN} actually.

Table 1: The methods used in this comparative evaluation

Notation	Algorithm	Parameter			
		μ	ρ	ϵ	δ
*ME-C	MAP-Elites	10^4	10^2	10^{-5}	10^2
*ME-LB				10^{-5}	0
*ME-GW				0	10^2
ME-I				0	0
GA-C	EGA	10^4	10^2	10^{-5}	10^2
GA-I				0	0
NR	RW-BFS [18]	-	-	-	-

Table 2: Parameter settings of the substrate network

	Node Type					Link Type		
	V_U	V_{NB}	V_{EC}	V_{CC}		$E_{U,NB}$	$E_{NB,EC}$	$E_{EC,CC}$
$\|V\|$	90	30	10	1	$\|E\|$	$[1,3]$	$[2,6]$	$[1,1]$
C, M	-	$[100,200]$	$[200,700]$	$[5000,10000]$	T	$[50,80]$	$[80,150]$	$[200,500]$
					L	$[3,7]$	$[3,5]$	$[2,4]$

Evaluation Metrics For the quantification of the degree of the total optimization, we define the acceptance ratio of network slice requests as the metric.

At the time t, the acceptance ratio is given by

$$\frac{|\{N_R \in \mathcal{R} \mid \exists t\,(M_A(N_R \mid t) = 1)\}|}{|\mathcal{R}|}$$

Furthermore, the profitability is also an important metric to evaluate NSE methods, therefore we define the revenue and the revenue to cost ratio (R/C ratio) like the previous work [18]. The revenue $Rv(t)$ and the cost $Cs(t)$ are defined as

$$Rv(t) := \sum_{k \in \mathbb{N}_A} \left\{ \sum_{X \in \{C,M\}} \sum_{a \in V_A^k} X(a) + \sum_{l \in E_R^k} T(l) \right\}$$

$$Cs(t) := \sum_{k \in \mathbb{N}_A} \left\{ \sum_{X \in \{C,M\}} \sum_{a \in V_A^k} X(a) + \sum_{l \in E_R^k} (|M_L^k(l)| - 1) T(l) \right\}$$

Consequently, the R/C ratio is given by

$$\frac{Rv(t)}{Cs(t)}$$

In addition, the average revenue over the entire duration is called the long-term revenue, which is given by

$$\lim_{T \to \infty} \frac{\sum_{t=0}^{T} Rv(t)}{T}$$

Similarly, the long-term R/C ratio can also be defined as

$$\lim_{T \to \infty} \frac{\sum_{t=0}^{T} Rv(t)}{\sum_{t=0}^{T} Cs(t)}$$

5.1 Comparison under the Dynamic Environment

For the performance comparison under the realistic dynamic environment, NSE using each method is simulated. The parameter settings about network slice requests in this simulation are shown in Table 3. The same network slice requests and arrival times are used in every simulation. For the methods except for NR, the simulations are performed 30 times in the same settings since genetic algorithms are probabilistic.

The simulation results for each method are shown in Table 4 and Figure 5–8. Table 4 shows that our proposed method ME-C accepts the most slices and produces the most

Table 3: The network slice requirements by service types in scenario 1

	Service Type		
	URLLC	eMBB	mMTC
$\lvert V'_U \rvert$	$[1, 10]$	$[1, 10]$	$[15, 30]$
$\lvert V_A \rvert$	$[1, 5]$	$[1, 10]$	$[1, 5]$
C, M	$[3, 15]$	$[10, 40]$	$[1, 3]$
T	$[5, 15]$	$[10, 20]$	$[1, 5]$
L	$[10, 30]$	$[25, 50]$	$[50, 100]$

long-term revenue, which reveals that the total optimization is realized by the cooperative behavior in our methods. The metrics of all methods using MAP-Elites are larger than those of other methods. This means that MAP-Elites searches diverse feasible solutions most efficiently in larger search space. While the difference of the acceptance ratio among the methods using MAP-Elites is small, the difference of the long-term revenue is large relatively. Moreover, ME-LB accepts more slices than ME-GW, whereas ME-GW produces more revenue than ME-LB. This result indicates that the requests with more resource requirements tend to be accepted by the cooperative behavior "giving way". On the other hand, the R/C ratios of ME-C and ME-GW are less than the other ME, which is caused by the selection of longer paths. It is found from these result that the selection of detour paths for giving way results in the acceptance of the requests with more resource requirements.

Table 4: Evaluation metrics in scenario 1

	ME-C	ME-LB	ME-GW	ME-I	GA-C	GA-I	NR
Acceptance ratio(%)	65.5	65.2	64.8	64.4	62.3	61.2	51.3
Long-term revenue	2861.0	2755.0	2849.4	2709.3	2589.8	2478.3	2102.7
Long-term R/C ratio(%)	50.3	55.4	50.2	55.4	49.6	53.9	50.7

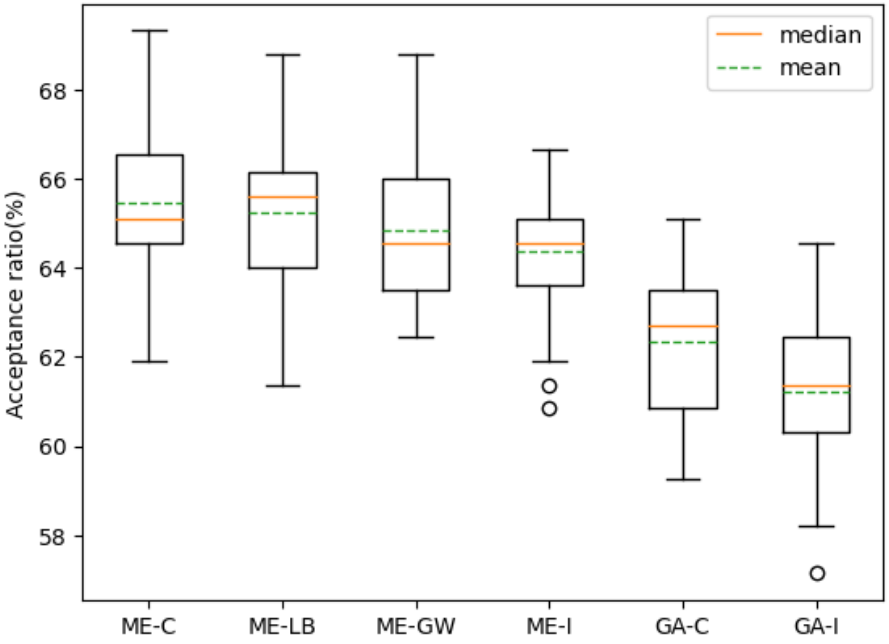

Figure 5: Acceptance ratio through a single simulation in scenario 1

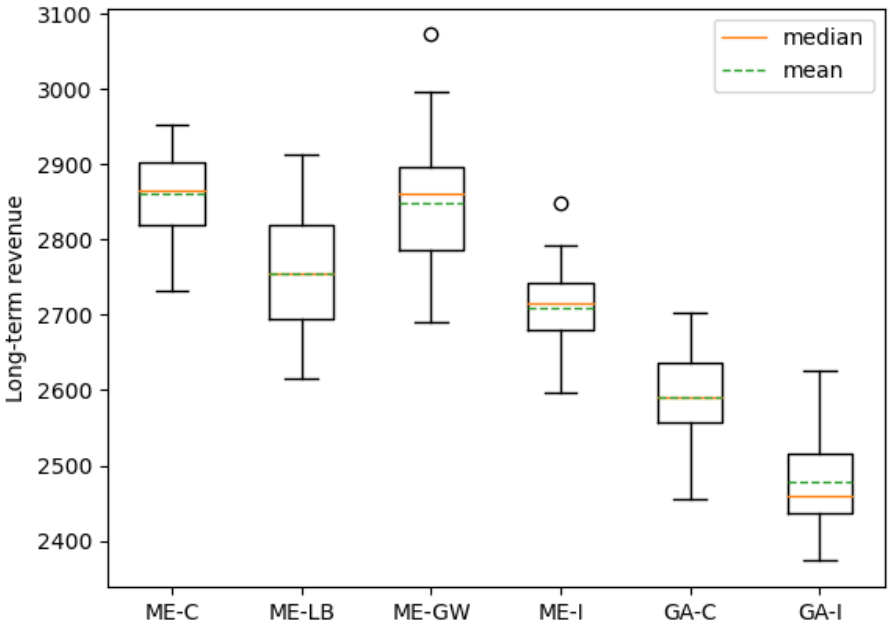

Figure 6: Long-term revenue through a single simulation in scenario 1

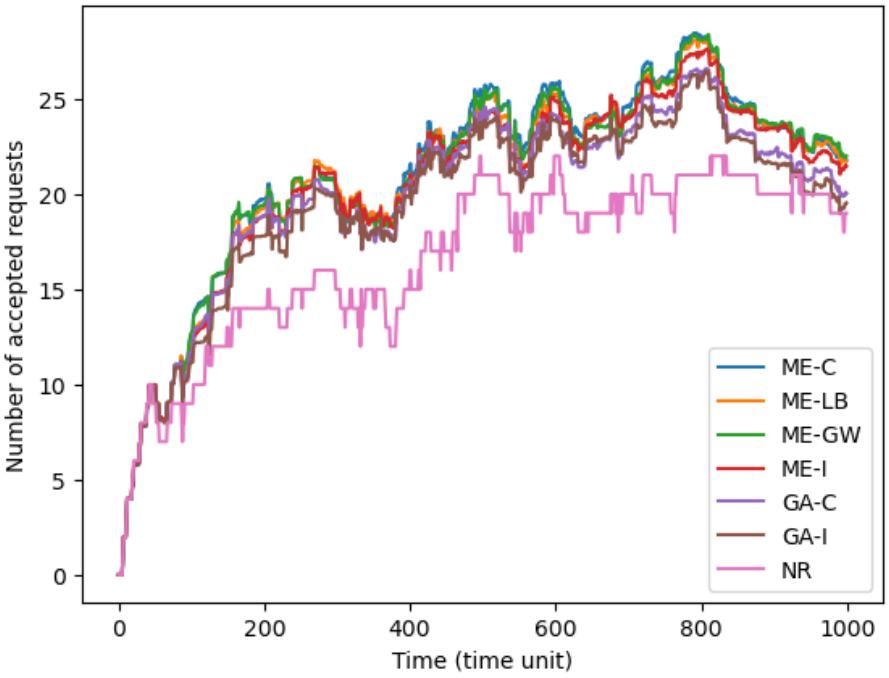

Figure 7: The transition of the number of accepted network slices over time in scenario 1

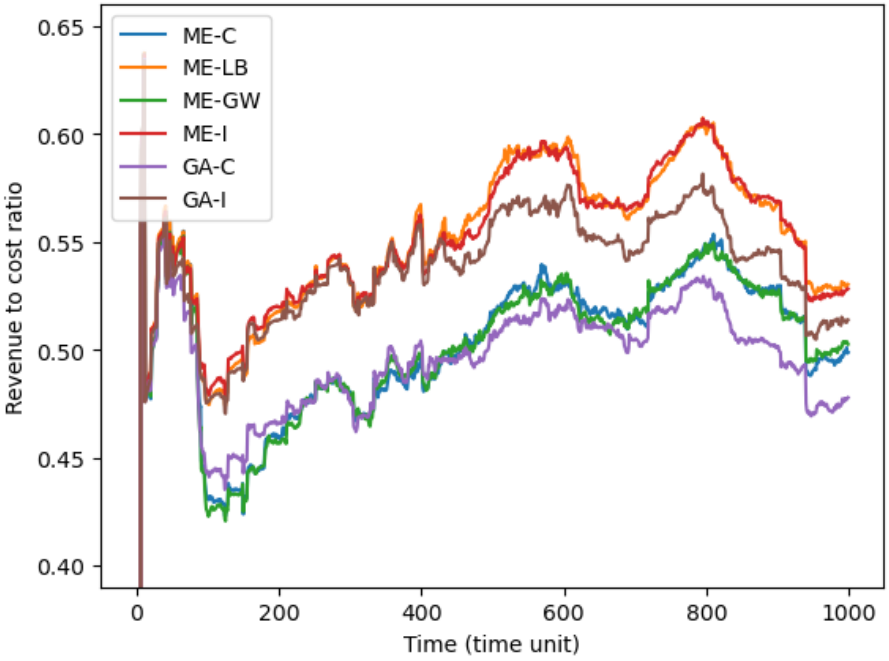

Figure 8: The transition of R/C ratio over time in scenario 1

5.2 Comparison under the Static Environment

For the comparison the profits among the proposed method, we evaluate the methods using MAP-Elites in scenario 2. The parameter settings about network slice requests in this simulation are shown in Table 5.

Table 5: The network slice requirements by service types in scenario 2

	Service Type		
	URLLC	eMBB	mMTC
$\lvert V'_U \rvert$	$[1, 5]$	$[1, 10]$	$[15, 30]$
$\lvert V_A \rvert$	$[1, 3]$	$[1, 5]$	$[1, 5]$
C, M	$[3, 15]$	$[10, 40]$	$[5, 10]$
T	$[2, 3]$	$[3, 5]$	$[1, 2]$
L	$[16, 20]$	$[25, 50]$	$[50, 100]$

Simulations are conducted for each of three the sequence of slice requests generated based on these settings. The transition of number of accepted requests is depicted in Figure 9. According to this figure, the methods are classified into two groups; the group including ME-C and ME-GW, and the others. It is shown that the number of accepted requests of ME-C and ME-GW increases rapidly, which means that the search of feasible solutions using resources with less demands promotes the acceptance of the other slices. The more early acceptance of more slices is desirable, and the increase of acceptance ratio in the evaluation under scenario 1 should be caused by this effect. On the other hand, it depends on the case which method accept the most slices finally. This suggests that the properties of the network slice requests affect the effectiveness of the cooperative behaviors.

Based on these figures, it seems the difference between ME-C and ME-LB is negatively correlated with the final number of accepted requests, which implies that the advantage of giving way becomes large if the substrate resources are abundant, and the advantage of load balancing becomes large if not. However, since no statistic analysis is conducted about this relation, the research about the relation between the properties of requests and the effectiveness of methods is feature work.

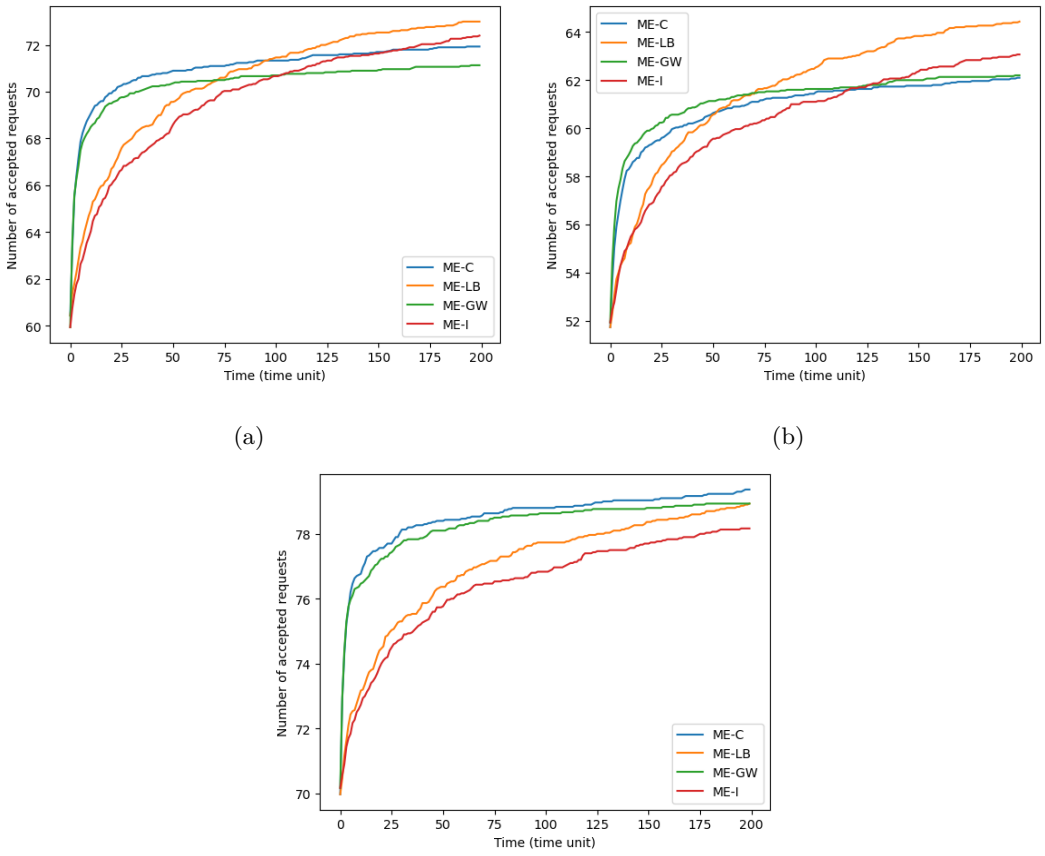

Figure 9: The transition of the number of accepted requests over time in scenario 2

6 Conclusion

For the dynamic NSE problem in which it is needed to deal with multiple network slices with diverse requirements, we proposed the total optimization method that optimizes each slice with individual controller behaving cooperatively in parallel. Taking advantage of the design flexibility of the objective function in QD algorithm, we incorporated the unselfish evaluation values into the fitness to realize cooperative behavior, which enables the total optimization by separate controllers. Through computer simulations under the realistic dynamic scenario and the static scenario, the comparative evaluation revealed the effectiveness of our proposed method and the properties of each variant.

In the evaluation, the weighting parameters in the fitness are fixed to certain values, although we did not conduct the search for the appropriate setting of the parameters. For example, setting the parameter δ large caused of excess usage of substrate resources in the conducted simulations. As the finer parameter settings may lead to the improvement of the performance of proposed methods, this is future work.

Although we made the dedicated simulation program for the evaluation, a sufficient number of simulations cannot be conducted due to the execution time. For further detailed research, the simulation program must be redesigned to reduce of the execution time. The evaluation of the solution candidate can be formulated as matrix operation and accelerated by using GPU. Subsequently to the acceleration, the large scale simulation should be conducted to analyze the properties of the methods statistically. These are also future works.

Acknowledgments

For the completion of this thesis, I have been given a lot of great advice and insightful suggestion by Professor Masayuki Murata of the Graduate School of Information Science and Technology at Osaka University, and I appreciate it. In regular meetings, he sometimes gave me compliments, and that encouraged me. I also thank for his tolerance for my fault and his continuous direction.

I was also daily instructed and helped with the progress of my research by Associate Professor Daichi Kominami of the Institute for Open and Transdisciplinary Research Initiative, Osaka University. Thanks to his considerable assistance, I proceeded with my research appropriately and completed this thesis finally. I always depended on him for even the trivial daily things not related in the contents of research. I would like to express my greatest gratitude to him.

The subject of this thesis is affected directly by the previous work of Assistant Professor Tatsuya Otoshi of the Graduate School of Economics, Osaka University. In the early phase of my research, I asked him for many knowledge about the problem and the implementation. Without his cooperation, I would have much trouble advancing my research. I am most grateful to him.

I also would like to express my gratitude for giving me invaluable advice to Associate Professor Shin'ichi Arakawa and Specially Appointed Assistant Professor Masaaki Yamauchi of the Graduate School of Information Science and Technology, Osaka University.

Finally, I would like to thank all the members of the Advanced Network Architecture Research Laboratory.

References

[1] F. Debbabi, R. Jmal, L. C. Fourati, and A. Ksentini, "Algorithmics and modeling aspects of network slicing in 5G and beyonds network: Survey," *IEEE Access*, vol. 8, pp. 162 748–162 762, 2020.

[2] K. Ludwig, A. Fendt, and B. Bauer, "An efficient online heuristic for mobile network slice embedding," in *Proceedings of Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, pp. 139–143.

[3] A. Fendt, C. Mannweiler, L. C. Schmelz, and B. Bauer, "A formal optimization model for 5G mobile network slice resource allocation," in *Proceedings of IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2018, pp. 101–106.

[4] C. Jiang and P. Zhang, *Virtual Network Embedding Based on Modified Genetic Algorithm*. Springer Singapore, 2021, pp. 109–127.

[5] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2009, pp. 783–791.

[6] H. Cao, H. Hu, Z. Qu, and L. Yang, "Heuristic solutions of virtual network embedding: A survey," *China Communications*, vol. 15, no. 3, pp. 186–219, 2018.

[7] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," 2015.

[8] A. Otsuki, D. Kominami, T. Otoshi, H. Shimonishi, and M. Murata, "Adaptive network slicing control method for unpredictable network variations using quality-diversity algorithms," in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC)*, 2024.

[9] 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*. Association for Computing Machinery, 2016, p. 173–180.

[10] NGMN, *NGMN 5G white paper 2*. NGMN Alliance, Jul. 2020.

[11] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The algorithmic aspects of network slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.

[12] H.-T. Chien, Y.-D. Lin, C.-L. Lai, and C.-T. Wang, "End-to-end slicing as a service with computing and communication resource allocation for multi-tenant 5G systems," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 104–112, 2019.

[13] K. Chatzilygeroudis, A. Cully, V. Vassiliades, and J.-B. Mouret, "Quality-diversity optimization: a novel branch of stochastic optimization," 2020.

[14] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011.

[15] ——, "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.

[16] 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.

[17] I. Labrador Pavon, J. (Atos, A. Colazzo, R. Ferrari, P. Arnold, M. Breitbach, H. Droste, D. Telekom, V. Friderikos, S. London, V. Sciancalepore, Z. (NEC, M.-L. Alberi-Morel, M. Doll, B. Gajic, S. Kerboeuf, C. Mannweiler, D. Michalopoulos, P. Rost, and A. Madrid, "5G NORMA network architecture – intermediate report," Jan. 2017.

[18] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, p. 38–47, Apr. 2011.