Virtualized Network Graph Design and Embedding Model to Minimize Provisioning Cost

Takehiro Sato, *Member, IEEE,* Takashi Kurimoto, *Member, IEEE,* Shigeo Urushidani, *Member, IEEE,* and Eiji Oki, *Fellow, IEEE*

***Abstract*—The provisioning cost of a virtualized network (VN) depends on several factors, including the numbers of virtual routers (VRs) and virtual links (VLs), mapping of them on a substrate infrastructure, and routing of data traffic. An existing model, known as the virtual network embedding (VNE) model, determines the embedding of given VN graphs into the substrate infrastructure. When the resource allocation model of the VNE problem is adopted to a single-entity scenario, where a single entity fulfills the roles of both a service provider and an infrastructure provider, an issue of increased costs of VNs and access paths arise. This paper proposes a model for virtualized network graph design and embedding (VNDE) for the single-entity scenario. The VNDE model determines the number of VRs and a VN graph for each request in conjunction with embedding. The VNDE model also determines access paths that connect customer premises and VRs. We formulate the VNDE model as an integer linear programming (ILP) problem. We develop heuristic algorithms for the cases where the ILP problem cannot be solved in practical time. We evaluate the performance of the VNDE model on several networks, including an actual Japanese academic backbone network. Numerical results show that the proposed model designs suitable VN graphs and embeds them according to the volume of traffic demands and access path cost. Compared with the benchmark model, which is based on a classic VNE approach, the proposed model reduces the provisioning cost at most 28.7% in our examined scenarios.**

***Index Terms*—Network virtualization, optimization model, integer linear programming, provisioning cost.**

I. Introduction

NETWORK virtualization technologies, such as software-defined networking (SDN) [1] and network function virtualization (NFV) [2], [3], enable network operators to control their network and computation resources in a flexible manner. A network operator provides their customers with virtualized networks (VNs); the VNs connect customer premises to requested services and are configured on the operator's infrastructure. Developments in automated network management technologies [4]–[6] can reduce the effort of manual network configuration and achieve on-demand provisioning of VNs.

A VN is provisioned by deploying virtual routers (VRs) and establishing virtual links (VLs) between VRs on a substrate infrastructure. Each of the customer premises, such as data centers (DCs), connects to one of the VRs. Data can be exchanged among customer premises via VRs. VR-based VN provisioning allows data from each premise to be aggregated at a VR and forwarded to their destinations; it leads to efficient use of network resources. The provisioning cost of a VN depends on several factors, including the number of VRs, connections between VRs provided by VLs, mapping of VRs and VLs on the substrate infrastructure, and routing of data traffic exchanged between customer premises. The network operator needs to design VNs and embed them into the substrate infrastructure to provide their customers with cost-efficient VNs.

The virtual network embedding (VNE) problem has been widely studied [7]–[13]. The VNE problem determines the mapping of VN requests, each of which is expressed by a network graph, onto the substrate infrastructure. Transmission capacity and computation resources of the substrate infrastructure are allocated to VLs and VRs, respectively, to satisfy the VN requests. The VNE problem considers scenarios where service providers (SP) and infrastructure providers (InP) have different roles in the VN provisioning [7], [8]. For example, the SP first designs VNs and then requests the InP to allocate infrastructure resources to them [12], [13]. The InP determines the resource allocation based on the SP's request. This scenario has the advantage that the SP is free of the burden of deploying and managing the substrate infrastructure and thus is able to focus on providing services.

There is another scenario in which a single entity fulfills the roles of both SP and InP[1]. In this single-entity scenario, the single entity can design and embed VNs by considering the conditions of the substrate infrastructure in detail, such as the utilization rate and cost of substrate routers and links.

When the resource allocation model of the VNE problem is adopted to the single-entity scenario, the following issues arise. First, VN design is separated from VN embedding, since the VNE problem only determines the embedding of given network graphs of VN requests. This can prevent customers from realizing cost-effective network designs for their VNs. Second, the VNE problem does not consider the cost of access paths between customer premises and VRs. This can trigger the improper placement of VRs, which imposes unnecessary

This work was supported in part by ROIS NII Open Collaborative Research 20S0104, and JSPS KAKENHI Grant Numbers 18H03230 and 19K14980.

T. Sato and E. Oki are with Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606–8501 Japan (e-mail: takehiro.sato@i.kyoto-u.ac.jp).

T. Kurimoto and S. Urushidani are with Information Systems Architecture Science Research Division, National Institute of Informatics, 2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430 Japan.

[1]For example, the National Institute of Informatics provides a Japanese academic backbone network called the Science Information Network 5 (SINET5) [14]. SINET5 offers various virtual network services, including layer-2 and layer-3 virtual private network services, virtual private local area network service, layer-2 on-demand service, and SDN services [15].

cost on customers. A model that addresses the above two issues needs to be developed.

This paper proposes, for the single-entity scenario, a model for virtualized network graph design and embedding (VNDE). This paper addresses an offline problem; the VNDE model determines the number of VRs, a VN graph, and its embedding for each VN request based on given traffic demands between every pair of customer premises. The VNDE model also determines the access paths between customer premises and VRs. We formulate the VNDE model as an ILP problem, in which the utilization costs of VRs and VLs are provided as non-decreasing step functions. The objective function of the ILP problem is to minimize the cost required for provisioning all VN requests. We develop heuristic algorithms for the cases in which the ILP problem cannot be solved in practical time. We evaluate the performance of the VNDE model by applying it to several networks, including real-world ones such as SINET5. Numerical results show that the proposed model designs VN graphs and embeds them into the substrate infrastructure according to the volume of traffic demands and access path costs; the provisioning cost can be suppressed by deploying multiple VRs when the volume of traffic demands and the utilization cost of substrate links for access paths become large.

A part of this paper was presented in [16]. The extensions to the work in [16] are mainly described as follows. We provide a detailed description of the ILP formulation of the VNDE model, including cost functions and constraints. We introduce heuristic algorithms for the VNDE model in case that the ILP problem is intractable, i.e., the computation time required for obtaining a solution exceeds the time period available to the network operator. We provide numerical results with various parameter values, which include the actual parameters of SINET5. The performance of VNDE model is compared with a benchmark model, which determines the VN graph design and the VN embedding step-by-step. We extensively survey existing research related to our work.

The rest of the paper is organized as follows. Section II describes related work. Section III presents the optimization model of VNDE. Section IV presents heuristic algorithms for the VNDE model. Section V presents the benchmark algorithm used in performance evaluations in this paper. Section VI shows numerical results of VN provisioning with the proposed model. Section VII concludes this paper.

II. Related work

The problem of mapping virtual resources of VNs to the substrate network is known as the VNE problem, and is extensively surveyed in [7]–[11]. The VNE problem determines the embedding of network graphs of VNs upon a substrate network. On the other hand, the VNDE problem studied in this paper determines the design of VN for each VN request in addition to its embedding. The VN design aims to find the number of VRs and connections between them so that the VN embedding can be achieved cost-efficiently. Fischer et al. [8] categorized the VNE approaches in the literature according to three aspects: whether the VN control is *centralized* or *distributed*, whether VN requests and substrate infrastructure change over time (*static* or *dynamic*), and whether there is redundancy in resource allocation for fault tolerance (*concise* or *redundant*). Following this taxonomy, the optimization model presented in Section III can be classified as a centralized, static, and concise approach.

Various types of objective functions are utilized in existing studies on the VNE problem [7], [8]. There are, for example, objective functions that aim to balance loads on substrate resources [17]–[19], maximize the revenue of SPs and InPs [20]–[24], maximize the acceptance ratio [22]–[25], minimize the energy consumption [26]–[29], and minimize the provisioning cost of VNs [30]–[32]. Some of the studies define the objective function as a weighted sum of several different metrics to achieve multiple goals [33]–[39]. The VNDE problem sets the provisioning cost as the objective function, but the definition of the cost is different from the existing VNE studies. The VNDE problem handles the utilization costs of VRs and VLs, which are given as non-decreasing step functions, and the cost for establishing access paths.

There have been several works that use traffic matrixes in studying the VNE problem [22], [30], [33]. Wang et al. [30] pointed out that a VNE approach with given VN graphs can limit the substrate network utility, and provided a formulation of the VN mapping problem based on a traffic matrix. Ivaturi et al. [33] expanded the approach in [30] so that VN mapping is determined with consideration of the end-to-end delay. In [30] and [33], the number of hub nodes, which correspond to VRs in the VNDE problem, is fixed; it needs to be chosen from among several values to improve the substrate network utility. Wen et al. [22] worked on a VNE problem that allows computation resource sharing between low-priority virtual nodes and the partial acceptance of VN requests. The number of virtual nodes is given in each VN request and allowed to be reduced only when there is a shortfall in substrate resources. On the other hand, the VNDE problem studied in this paper determines the number of VRs so as to suppress the provisioning cost of VNs based on a given traffic matrix.

There have been existing works on the VNE problem in a dynamic scenario [8]. In the dynamic scenario, VN requests are not known in advance; they discretely arrive and hold in a network system for a certain period of time. Generally, an online VNE algorithm is required to determine the allocation of substrate resources in a short time since it directly impacts on waiting time for users requesting a VN. Aguilar-Fuster et al. [40] presented an online VNE approach that is aware of the computation resources required by each node on VLs to forward traffic. The authors implemented their approach by using metaheuristic techniques: particle swarm optimization, genetic algorithm, and harmony search. The performance evaluation was carried out by using 50-node and 100-node substrate networks. Dehury et al. [41] developed a dynamic VNE algorithm based on fitness values. The algorithm allows reconfiguration of network resources in response to fluctuations in resource demand during VN operations. Cao et al. [42] developed a dynamic VNE algorithm that considers the quality-of-service (QoS) performance of each VN. When a VN request arrives, the VN is mapped to the substrate network so that the resource demands can be satisfied. Then, the embedded VN is adjusted in order to meet the QoS demand. The VN reconfiguration is another peculiar problem in the dynamic scenario; the mapping of existing VNs can be reconfigured to improve the resource utilization efficiency while serving VNs [43], [44]. VNs need to be updated so that they do not experience service interruptions due to link congestions, forwarding loops, and so on, during the reconfiguration process. This paper mainly focuses on the VNDE problem in a static scenario, where the accuracy of VN provisioning in terms of the provisioning cost is required. The proposed model can be adopted to the dynamic scenario by inputting newly-arrived VN requests and setting available resources as parameters. The model also can be utilized to determine the new mapping of existing VNs before the VN reconfiguration process.

Some recent works on the VNE problem [45]–[49] adopt machine learning techniques, especially reinforcement learning, to obtain VN embeddings that meet users' requests in a resource-efficient manner. The reinforcement learning technique has advantages in autonomously discovering hidden features of a network and VN requests and learning an efficient embedding policy. On the other hand, sufficient time is required for the learning; the learning agent may perform inefficient or unacceptable embeddings until the learning process has been progressed enough. The reinforcement learning technique can be adopted to the VNDE problem for the online operation, which is left for a future work.

Traffic grooming is another research area similar to the VNDE problem. It multiplexes several low-speed traffic flows into a high-capacity wavelength channel in optical networks, such as wavelength division multiplexing (WDM) networks [50], [51] and elastic optical networks (EONs) [52]. Network operators can improve the utilization efficiency of wavelength resources and suppress the cost of optical network equipment, such as optical transceivers and wavelength converters, by adopting traffic grooming. Dutta et al. [50] defined the traffic grooming problem as follows; given a traffic matrix on a network, the problem determines a set of lightpath requests, their routing and wavelength assignment, and routing of traffic flows on the lightpaths. The goal of the problem is to minimize a cost function related to lightpath establishment, e.g., the total number of lightpaths. The traffic grooming problem and the VNDE problem may be similar in that they aggregate traffic flows at a node (i.e., an optical cross connect (OXC) or a VR) and transfer an aggregated traffic flow on a path (i.e., a lightpath or a VL). The major difference between the two problems is that, in the VNDE problem, access paths are computed as well as VLs so that the total cost imposed on customers can be minimized.

The VNDE problem studied in this paper is related to the VNE problem, and both problems aim to determine the allocation of substrate network resources to VN requests. The major difference between them is that, in the VNDE problem, VN design is performed concurrently with VN embedding, which leads to cost-efficient VN deployment for network operators and customers. The VNDE model selects the suitable number of VRs and arrangement of VLs that suppress the total provisioning cost. The placement of VRs is determined so that the access path cost, which is imposed on customers, can be discounted. The VNDE model is capable of fine-tuned cost settings of VRs and VLs by giving them as step functions.

Table I summarizes the related works on the VNE problem introduced in this section.

III. Optimization model

A. Overview

Fig. 1 shows an example of VN graph design and embedding considered in this paper. Fig. 1(a) presents a physical connection configuration. A substrate network, which is composed of substrate routers and substrate links, is available for provisioning VNs and establishing access paths. VRs can be hosted as software in each substrate router by using the computation resource that the substrate router has. Customer premises, i.e., DCs, are connected to the substrate routers. Note that DCs, including internal links that connect the DCs to the substrate routers, are distinct from the substrate network. Customer traffic from a source DC to a destination DC can pass through a substrate router where another DC is attached, but the traffic does not traverse the DC itself. A part of substrate routers are capable of hosting VRs. The VNDE model considers logical connections between the VR-capable routers, as shown in Fig. 1(b). A VL that connects a pair of VRs can be established along a logical connection. VRs and VLs make up a VN provided to the customer. An access path is established between a DC and a VR so that each DC can access a VN and exchange data with other DCs. If a substrate router to which the DC is attached and that hosting the corresponding VR are different, an access path is set up by using substrate links. Fig. 1(c) shows possible solutions of resultant VN and access paths when a customer has three DCs and the maximum number of VRs is set to three. The solution enclosed by a blue rectangle corresponds to Fig. 1(b).

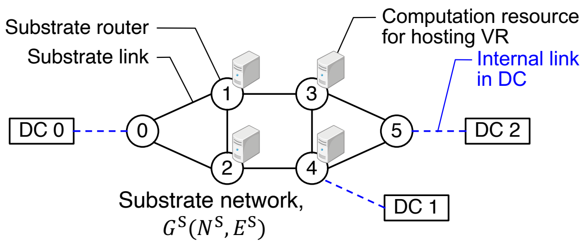
A customer's VN request is mainly characterized by locations of DCs, the traffic demand of each source-destination DC pair, and the maximum allowable delay for each source-destination DC pair. Capacities of substrate routers and substrate links are allocated to VN requests to deploy VRs, VLs, and access paths. A unidirectional unicast flow, which passes through an upstream access path, VRs, VLs, and a downstream access path, is determined for each source-destination DC pair. A VN is constructed so that all the unicast flows can be transferred by satisfying the traffic demand and the delay requirement. The VNDE model attempts to find the optimum VN design and embedding for all VN requests at once.

In Fig. 1, substrate routers 1, 2, 3, and 4 are capable of hosting VRs. Logical connections are provided between these routers in a full-meshed manner. A customer has three DCs, which are located on substrate routers 0, 4, and 5. VRs are placed on routers 2 and 4, and a VL is set up between them. Substrate links (0, 2) and (2, 0) (expressed by one link in Fig. 1 for simplicity) are used for an upstream and downstream access path, respectively, so that DC 0 can access a VR on router 2. In the same way, substrate links (5, 4) and (4, 5) are used for access paths so that DC 2 can access to a VR on router 4.

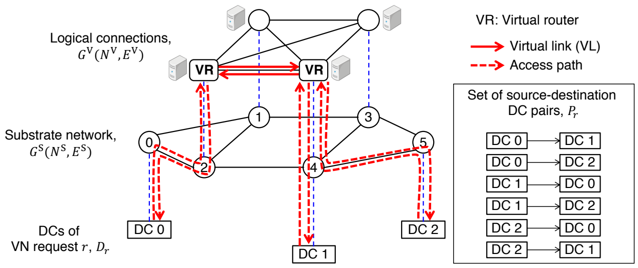
TABLE I
SUMMARY OF RELATED WORKS ON VNE PROBLEM.

Work	Approach	Objective	Contribution
Zhu et al. [17]	Heuristic	Minimizing maximum resource usage of substrate nodes and links	Algorithms for VN assignment problem with/without VN reconfiguration
Melo et al. [18]	Optimization	Minimizing maximum resource usage of substrate nodes and links	ILP formulation for online VNE
Mijumbi et al. [19]	Column generation	Minimizing weighted resource usage for achieving load balancing	Column generation-based approach for one-shot embedding (virtual nodes and links are embedded in one step)
Yu et al. [20]	Heuristic	Maximizing long-term average revenue	Algorithm that adopts path splitting and periodical path migration
Zhang et al. [21]	Optimization and heuristic	Maximizing revenue of InP	Framework where substrate resources are shared among VNs in time-division manner
Wen et al. [22]	Heuristic	Maximizing long-term average revenue and acceptance ratio	VN mapping algorithm that allows node sharing and partial acceptance of VN requests
Zhang et al. [23]	Heuristic	Maximizing scaling factor of VLs, revenue/cost ratio, and acceptance ratio	Algorithm based on node degree and clustering coefficient
Gong et al. [24]	Heuristic	Maximizing revenue of InP by accommodating as many VNs as possible	Algorithms for location-constrained VNE based on compatibility graph
He et al. [25]	Optimization and heuristic	Maximizing number of VN requests successfully embedded	Mixed-integer linear programming (MILP) model and heuristic algorithm for VNE in IoT network environment
Botero et al. [26]	Optimization	Minimizing energy consumption by switching off as many network nodes and links as possible	MILP model for energy-aware VNE problem
Su et al. [27]	Optimization, heuristic, metaheuristic	Minimizing node, link, and switching energy costs	ILP model and algorithms for energy-aware VNE problem
Nonde et al. [28]	Optimization and heuristic	Minimizing overall power consumption in network and data centers	MILP model and heuristic algorithm for VNE in IP-over-WDM networks with data centers
Chen et al. [29]	Optimization and heuristic	Minimizing overall energy cost	Optimization model and heuristic algorithm for energy efficient VNE
Wang et al. [30]	Optimization	Minimizing total cost of VN mapping	Mixed-integer programming (MIP) model of VN mapping problem based on traffic matrices
Lin et al. [31]	Optimization and heuristic	Minimizing embedding cost in terms of spectrum slot cost and computation cost	ILP formulation and heuristic algorithms for VNE problem with geographical constraints in flexi-grid optical networks
Chowdhury et al. [32]	Optimization and heuristic	Minimizing cost for IP links and VL bandwidth	ILP formulation and heuristic algorithm for VNE problem in multi-layer IP-over-optical networks
Ivaturi et al. [33]	Optimization	Minimizing weighted function consisting of resource utilization and sensitivity to end-to-end delays	MIP model of VN mapping that considers end-to-end delay
Chowdhury et al. [34]	Optimization and heuristic	Minimizing VN embedding cost, along with load balancing	Online VNE algorithms based on solving linear programming (LP)-relaxed problem
Cao et al. [35]	Optimization and heuristic	Minimizing cost or link propagation delay, along with load balancing	VNE algorithm that suppresses computational complexity by constructing substrate candidate subsets and then running ILP-based mapping
Li et al. [36]	Self-adaptive algorithm	Maximizing acceptance ratio and revenue in long term	Self-adaptive VNE algorithm based on ILP for multi-demand problem of tenants
Chai et al. [37]	Optimization and heuristic	Minimizing network load and maximizing embedding reliability	Multi-objective optimization problem and algorithm for VNE in SDN where substrate resources can be maliciously attacked
Phan et al. [38]	Optimization and heuristic	Minimizing cost and energy, along with avoiding network congestion	Multi-objective optimization model and heuristic algorithm for congestion- and energy-aware VNE problem in SDN
Lu et al. [39]	Optimization and heuristic	Minimizing resource utilization cost and fragmentation degree	MIP model and online VNE algorithm for resource fragmentation-aware VNE
Aguilar-Fuster et al. [40]	Metaheuristic	Minimizing amount of allocated CPU and bandwidth resources	Online VNE approach aware of computation resources to forward traffic
Dehury et al. [41]	Heuristic	Maximizing resource utilization by taking local and global fitness values	Dynamic VNE algorithm based on fitness values
Cao et al. [42]	Heuristic	Maximizing acceptance ratio while guaranteeing QoS performance of each accepted VN	Dynamic VNE algorithm that considers QoS performance of VN
Fajjari et al. [43]	Heuristic	Minimizing number of overloaded substrate links to reduce rejection of VNs	VN reconfiguration algorithm to minimize number of overloaded substrate links
Fan et al. [44]	Heuristic	Minimizing long-run average of overall cost of using overlay	Heuristic methods for constructing reconfiguration policies in dynamic overlay topology reconfiguration
Yao et al. [45]	Reinforcement learning	Maximizing long-term average revenue-to-cost ratio	VNE scheme regarding node embedding of same request as time-series problem
Yan et al. [46]	Reinforcement learning	Maximizing reward function considering request acceptance, long-term revenue, load balance, and policy exploration	Algorithm for automatic VN embedding based on deep reinforcement learning
Andreoletti et al. [47]	Reinforcement learning	Minimizing embedding cost while guaranteeing privacy requirements	Privacy-preserving reinforcement learning algorithm to perform VNE over multi-domain infrastructure
Thakkar et al. [48]	Reinforcement learning	Maximizing mean percentage of resource utilization based on predictions	Reinforcement learning-based prediction models for multi-stage VNE problem in cloud data centers
Cheng et al. [49]	Reinforcement learning	Maximizing long-term acceptance ratio and revenue	Proactive VNE algorithm based on hierarchical reinforcement learning

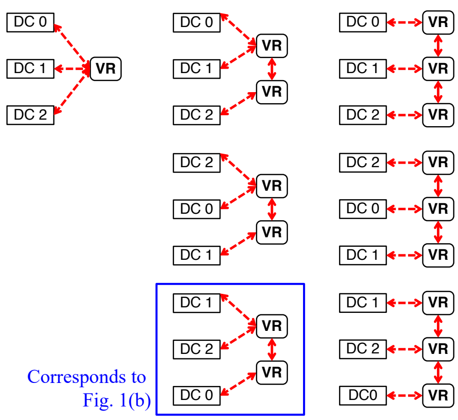
The VNDE problem aims to minimize the provisioning cost required to provide requested VNs. Different from the existing VNE studies, the VNDE problem distinguishes the utilization cost of a transit VR and that of an access VR. The provisioning cost in the VNDE problem includes the cost of using access paths. Fig. 2 shows the types of costs imposed in the VNDE model. The model considers three types of costs for provisioning VNs: costs of transit VRs, VLs, and access VRs. We call a VR *transit VR* when the VR transmits and receives traffic on VLs, and *access VR* when the VR is accessed by a DC to send data. Note that one VR can serve as both transit and access VRs at the same time, as the leftmost VR in Fig. 2 do. The costs of transit VRs, VLs, and access VRs are determined based on a non-decreasing step function. That



(a) Physical connection configuration.



(b) Virtual resource mapping.



(c) Possible solutions of resultant VN and access paths.

Fig. 1. Example of VNDE.

is, the model assumes a cost structure in which the utilization costs of VRs and VLs are determined based on classes, each of which has an upper limit on the amount of capacity used. Different cost functions can be set to the three types of costs. In addition to the above three costs, the access path cost is imposed on a customer when the customer sets up an access path from each of their DCs to the corresponding access VR. The model assumes that the access path cost is determined based on the utilization cost of substrate links.

The VNDE problem can be compatible with other objective functions used in the existing VNE studies, such as those introduced in Section II. In this paper, as one of the typical objective functions, we focus on the minimization of VN provisioning cost. The proposed model may need to be modified if other objective functions are adopted. For example, decision variables that represent the acceptance of each VN request need to be added when we maximize the acceptance ratio; the power consumption model of each substrate network component is required as a parameter when we minimize the total power consumption. In Section VI-A4, we demonstrate

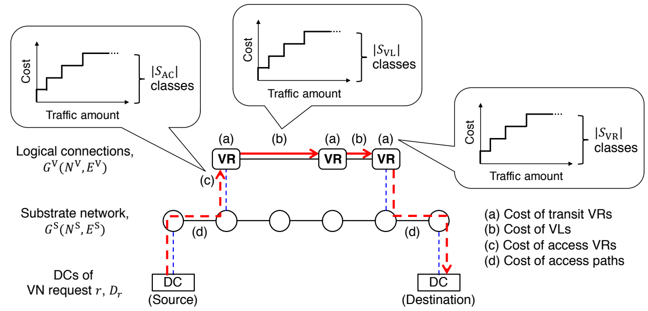


Fig. 2. Types of costs imposed in VNDE.

the performance of the proposed model when we use the load-balancing function as the objective function.

B. Terminologies and assumptions

The substrate network is denoted by directed graph $G^{\mathrm{S}}(N^{\mathrm{S}}, E^{\mathrm{S}})$, where N^{S} is a set of substrate routers and E^{S} is a set of substrate links. $(i, j) \in E^{\mathrm{S}}$ denotes an substrate link from $i \in N^{\mathrm{S}}$ to $j \in N^{\mathrm{S}} \backslash \{i\}$. c^{L}_{ij}, which is a given parameter, denotes the capacity of $(i, j) \in E^{\mathrm{S}}$.

R denotes a set of VN requests that are accommodated in $G^{\mathrm{S}}(N^{\mathrm{S}}, E^{\mathrm{S}})$. We introduce directed graph $G^{\mathrm{V}}(N^{\mathrm{V}}, E^{\mathrm{V}})$ to represent logical connections between routers. $N^{\mathrm{V}} \subseteq N^{\mathrm{S}}$ is a set of routers, each of which can be a VR. c^{N}_v, which is a given parameter, denotes the capacity of $v \in N^{\mathrm{V}}$. E^{V} is a set of logical links, each of which can be a VL. $(v, w) \in E^{\mathrm{V}}$ denotes a logical link that can be a VL from $v \in N^{\mathrm{V}}$ to $w \in N^{\mathrm{V}} \backslash \{v\}$ and is served by a series of substrate links in E^{S}.

D_r denotes a set of customer premises that belong to $r \in R$, each of which is directly connected to one of the substrate routers in N^{S}. In the following, a customer premise is simply referred to as a "DC." a_{rpi}, which is a given parameter, denotes a flag that is set to one if $p \in D_r$ is directly connected to $i \in N^{\mathrm{S}}$, and zero otherwise. As mentioned in Section III-A, DCs are considered to be separated from the substrate network in the VNDE model. If we treat DCs as a part of N^{S}, internal links that connect the DCs to the substrate routers need to be added to E^{S}. This leads to increase in the numbers of parameters and decision variables, which makes the model complicated.

Customer traffic from each DC passes through at least one VR when it is forwarded. DC $p \in D_r$ accesses a VR in $v \in N^{\mathrm{V}}$ through an access path, which can be set up by using a series of substrate links in E^{S}. DCs in D_r exchange their traffic each other via access paths and the VN configured for VN request $r \in R$. $(p, q) \in P_r$ denotes a pair of source DC $p \in D_r$ and destination DC $q \in D_r \backslash \{p\}$, where P_r denotes a set of source-destination DC pairs. d_{rpq}, which is a given parameter, denotes the traffic demand of $(p, q) \in P_r$ in VN request $r \in R$.

M_r, which is a given parameter, denotes the maximum allowable number of VRs in VN request $r \in R$. κ_{rpq}, which is a given parameter, denotes the maximum allowable delay for source-destination DC pair $(p, q) \in P_r$ in $r \in R$. β_v, which is a given parameter, denotes the delay at a VR placed on $v \in N^{\mathrm{V}}$.

TABLE II
SETS USED IN VNDE MODEL.

Symbols	Descriptions
R	Set of VN requests.
$G^{\mathrm{S}}(N^{\mathrm{S}}, E^{\mathrm{S}})$	A substrate network.
N^{S}	Set of substrate routers.
E^{S}	Set of substrate links.
$G^{\mathrm{V}}(N^{\mathrm{V}}, E^{\mathrm{V}})$	Logical connections between VR-capable routers.
N^{V}	Set of routers, each of which can be a VR. $N^{\mathrm{V}} \subseteq N^{\mathrm{S}}$.
E^{V}	Set of logical links, each of which can be a VL.
D_r	Set of customer premises (DCs) that belong to VN request $r \in R$.
P_r	Set of source-destination DC pairs in VN request $r \in R$.
K	Set of cost-function types. $K = \{\mathrm{VR}, \mathrm{VL}, \mathrm{AC}\}$.
S_k	Set of cost classes for cost-function type $k \in K$.

γ_{ij}, which is a given parameter, denotes the delay of substrate link $(i, j) \in E^{\mathrm{S}}$.

x_{rv} denotes a binary decision variable that is set to one if $v \in N^{\mathrm{V}}$ serves a VR in VN request $r \in R$, and zero otherwise. o_v^{rpq} denotes a binary decision variable that is set to one if the traffic of source-destination DC pair $(p, q) \in P_r$ passes through $v \in N^{\mathrm{V}}$ as a VR in $r \in R$, and zero otherwise. y_{rvw} denotes a binary decision variable that is set to one if logical link $(v, w) \in E^{\mathrm{V}}$ is used as a VL by $r \in R$, and zero otherwise. We assume that a VL in $r \in R$ is not separated into multiple routes on $G^{\mathrm{S}}(N^{\mathrm{S}}, E^{\mathrm{S}})$. λ_{ij}^{rvw} denotes a binary decision variable that is set to one if $(v, w) \in E^{\mathrm{V}}$ used by $r \in R$ passes through substrate link $(i, j) \in E^{\mathrm{S}}$, and zero otherwise. z_{rpv} denotes a binary decision variable that is set to one if DC $p \in D_r$ in $r \in R$ uses $v \in N^{\mathrm{V}}$ as an access VR, and zero otherwise. We assume that each DC accesses the same VR for both upstream and downstream directions, and traffic of $(p, q) \in P_r$ is not separated on multiple routes. $\theta_{ij,\mathrm{UP}}^{rpv}$ is a binary decision variable that represents the route of upstream access path; it is set to one if the upstream traffic from DC $p \in D_r$ in $r \in R$ to an access VR on $v \in N^{\mathrm{V}}$ passes through substrate link $(i, j) \in E^{\mathrm{S}}$, and zero otherwise. $\theta_{ij,\mathrm{DW}}^{rwq}$ is a binary decision variable that represents the route of downstream access path; it is set to one if the downstream traffic from an access VR on $w \in N^{\mathrm{V}}$ to $q \in D_r$ in $r \in R$ passes through $(i, j) \in E^{\mathrm{S}}$, and zero otherwise. ξ_{vw}^{rpq} denotes a binary decision variable that is set to one if the traffic of $(p, q) \in P_r$ in $r \in R$ passes through a VL configured on $(v, w) \in E^{\mathrm{V}}$, and zero otherwise.

Tables II, III, and IV summarize sets, parameters, and decision variables, respectively, used in the VNDE model.

C. Cost functions for virtualized networks

We define three types of cost functions, each of which is a function of traffic amount t. $f_{\mathrm{VR}}(t)$ is the usage cost of a VR as a transit VR. $f_{\mathrm{VL}}(t)$ is the usage cost of a VL between two VRs. $f_{\mathrm{AC}}(t)$ is the usage cost of a VR as an access VR. Let $K = \{\mathrm{VR}, \mathrm{VL}, \mathrm{AC}\}$ denote a set of cost-function types.

We assume that $f_k(t)$, where $k \in K$, is a non-decreasing step function. S_k denotes a set of cost classes for $k \in K$. In the definition of $f_k(t)$, if t is less than or equal to l_s^k, where $k \in K$ and $s \in S_k$, the cost is h_s^k. If $s_1 < s_2$, then $l_{s_1}^k < l_{s_2}^k$ and $h_{s_1}^k < h_{s_2}^k$. We set the minimum value of l_s^k over $s \in S_k$ to zero for all $k \in K$.

TABLE III
PARAMETERS USED IN VNDE MODEL.

Symbols	Descriptions
c_{ij}^{L}	Capacity of substrate link $(i, j) \in E^{\mathrm{S}}$.
c_v^{N}	Capacity of router $v \in N^{\mathrm{V}}$.
a_{rpi}	Flag that is set to one if DC $p \in D_r$ is directly connected to substrate router $i \in N^{\mathrm{S}}$, and zero otherwise.
d_{rpq}	Traffic demand of source-destination DC pair $(p, q) \in P_r$ in VN request $r \in R$.
M_r	Maximum allowable number of VRs in VN request $r \in R$.
κ_{rpq}	Maximum allowable delay for source-destination DC pair $(p, q) \in P_r$ in VN request $r \in R$.
β_v	Delay at a VR placed on $v \in N^{\mathrm{V}}$.
γ_{ij}	Delay of substrate link $(i, j) \in E^{\mathrm{S}}$.
l_s^k	Upper limit of traffic amount of cost class $s \in S_k$ for cost-function type $k \in K$.
h_s^k	Cost of cost-function type $k \in K$ when cost class is $s \in S_k$.
α_v^{VR}	Cost-coefficient parameter for transit VR usage cost at router $v \in N^{\mathrm{V}}$.
$\alpha_{vw}^{\mathrm{VL}}$	Cost-coefficient parameter for VL usage cost at logical link $(v, w) \in E^{\mathrm{V}}$.
$\alpha_{rpv}^{\mathrm{AC}}$	Cost-coefficient parameter for access VR usage cost at pair of DC $p \in D_r$ and router $v \in N^{\mathrm{V}}$.
ω_{ij}	Utilization cost of substrate link $(i, j) \in E^{\mathrm{S}}$ per unit transmission capacity.

TABLE IV
DECISION VARIABLES USED IN VNDE MODEL.

Symbols	Descriptions
x_{rv}	1 if router $v \in N^{\mathrm{V}}$ serves a VR in VN request $r \in R$, and 0 otherwise.
o_v^{rpq}	1 if the traffic of source-destination DC pair $(p, q) \in P_r$ passes through router $v \in N^{\mathrm{V}}$ as a VR in VN request $r \in R$, and 0 otherwise.
y_{rvw}	1 if logical link $(v, w) \in E^{\mathrm{V}}$ is used as a VL by VN request $r \in R$, and 0 otherwise.
λ_{ij}^{rvw}	1 if logical link $(v, w) \in E^{\mathrm{V}}$ used by VN request $r \in R$ passes through substrate link $(i, j) \in E^{\mathrm{S}}$, and 0 otherwise.
z_{rpv}	1 if DC $p \in D_r$ in VN request $r \in R$ uses router $v \in N^{\mathrm{V}}$ as an access VR, and 0 otherwise.
$\theta_{ij,\mathrm{UP}}^{rpv}$	1 if the upstream traffic from DC $p \in D_r$ in VN request $r \in R$ to an access VR on $v \in N^{\mathrm{V}}$ passes through substrate link $(i, j) \in E^{\mathrm{S}}$, and 0 otherwise.
$\theta_{ij,\mathrm{DW}}^{rwq}$	1 if the downstream traffic from an access VR on $w \in N^{\mathrm{V}}$ to DC $q \in D_r$ in VN request $r \in R$ passes through substrate link $(i, j) \in E^{\mathrm{S}}$, and 0 otherwise.
ξ_{vw}^{rpq}	1 if the traffic of source-destination DC pair $(p, q) \in P_r$ in VN request $r \in R$ passes through logical link $(v, w) \in E^{\mathrm{V}}$, and 0 otherwise.
μ_{vwij}^{rpq}	Equals to $\lambda_{ij}^{rvw} \xi_{vw}^{rpq}$.
$\sigma_{srv}^{\mathrm{VR}}$	1 if a VR served on router $v \in N^{\mathrm{V}}$ for VN request $r \in R$ imposes cost of class $s \in S_{\mathrm{VR}}$, and 0 otherwise.
$\sigma_{srvw}^{\mathrm{VL}}$	1 if a VL configured on $(v, w) \in E^{\mathrm{V}}$ is used by VN request $r \in R$ and imposes cost of class $s \in S_{\mathrm{VL}}$, and 0 otherwise.
$\sigma_{srvp}^{\mathrm{AC}}$	1 if router $v \in N^{\mathrm{V}}$ is used as an access VR of VN request $r \in R$ and imposes cost of class $s \in S_{\mathrm{AC}}$, and 0 otherwise.

The total cost of provisioning a VN for $r \in R$, C_r^{VN}, is given by:

$$C_r^{\mathrm{VN}} = \sum_{v \in N^{\mathrm{V}}} \alpha_v^{\mathrm{VR}}$$

$$f_{\mathrm{VR}}\left(\sum_{(p,q)\in P_r} d_{rpq}\left(\sum_{\substack{w\in N^{\mathrm{V}}:\\(v,w)\in E^{\mathrm{V}}}}\xi_{vw}^{rpq}+\sum_{\substack{w\in N^{\mathrm{V}}:\\(w,v)\in E^{\mathrm{V}}}}\xi_{wv}^{rpq}\right)\right)$$
$$+\sum_{(v,w)\in E^{\mathrm{V}}}\alpha_{vw}^{\mathrm{VL}} f_{\mathrm{VL}}\left(\sum_{(p,q)\in P_r} d_{rpq}\xi_{vw}^{rpq}\right)$$
$$+\sum_{v\in N^{\mathrm{V}}}\sum_{p\in D_r}\alpha_{rpv}^{\mathrm{AC}} f_{\mathrm{AC}}\left(z_{rpv}\sum_{q\in D_r:(p,q)\in P_r} d_{rpq}\right). \quad (1)$$

The first, second, and third terms of (1) represent the total costs of using transit VRs, VLs, and access VRs, respectively, in the network. α_v^{VR}, $\alpha_{vw}^{\mathrm{VL}}$, and $\alpha_{rpv}^{\mathrm{AC}}$ are given cost-coefficient parameters.

In addition to C_r^{VN}, the access path cost is imposed. It is expressed by:

$$\sum_{v\in N^{\mathrm{V}}}\sum_{(p,q)\in P_r} d_{rpq}\sum_{(i,j)\in E^{\mathrm{S}}}\omega_{ij}\left(\theta_{ij,\mathrm{UP}}^{rpv}+\theta_{ij,\mathrm{DW}}^{rvq}\right), \quad (2)$$

where ω_{ij} is a given parameter that denotes the utilization cost of substrate link $(i,j)\in E^{\mathrm{S}}$ per unit transmission capacity. Equation (2) means that the access path cost is proportional to the traffic amount and imposed for both upstream and downstream.

D. Constraints

We introduce the constraints of the VNDE model in this section. These constraints are a part of the ILP problem of the VNDE model, which is described in Section III-E.

1) Flow constraints: The flow constraints regarding a VL for traffic of source-destination DC pair $(p,q)\in P_r$ in VN request $r\in R$ are given by:

$$\sum_{w\in N^{\mathrm{V}}:(v,w)\in E^{\mathrm{V}}}\xi_{vw}^{rpq}-\sum_{w\in N^{\mathrm{V}}:(w,v)\in E^{\mathrm{V}}}\xi_{wv}^{rpq}=z_{rpv}-z_{rqv},$$
$$\forall r\in R,(p,q)\in P_r, v\in N^{\mathrm{V}}. \quad (3)$$

Equation (3) indicates that, if $v\in N^{\mathrm{V}}$ is an access VR for DC p and not for DC q in $(p,q)\in P_r$, i.e., if $z_{rpv}=1$ and $z_{rqv}=0$, the traffic of (p,q) is transmitted from v to any of logical link $(v,w)\in E^{\mathrm{V}}$. In this case, a VL is configured on $(v,w)\in E^{\mathrm{V}}$. On the contrary, if $v\in N^{\mathrm{V}}$ is an access VR for q and not for p in $(p,q)\in P_r$, i.e., if $z_{rqv}=1$ and $z_{rpv}=0$, v receives the traffic from any of $(w,v)\in E^{\mathrm{V}}$; a VL is configured on $(w,v)\in E^{\mathrm{V}}$. Equation (3) also guarantees the flow condition of intermediate VRs; the amounts of transmitted and received traffic are equal at a VR on v if both of z_{rpv} and z_{rqv} are zero. Note that, in case of $z_{rpv}=1$ and $z_{rqv}=1$, there is no traffic on any VL; (3) is satisfied with $\xi_{vw}^{rpq}=0$.

VLs need to be mapped on the substrate network. The flow constraints for the substrate route of $(v,w)\in E^{\mathrm{V}}$ are given by:

$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\lambda_{ij}^{rvw}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\lambda_{ji}^{rvw}=y_{rvw},$$
$$\text{if } i=v, \forall r\in R,(v,w)\in E^{\mathrm{V}}, \quad (4a)$$
$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\lambda_{ij}^{rvw}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\lambda_{ji}^{rvw}=0,$$
$$\text{if } i\neq v,w, \forall r\in R,(v,w)\in E^{\mathrm{V}}, \quad (4b)$$
$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\lambda_{ij}^{rvw}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\lambda_{ji}^{rvw}=-y_{rvw},$$
$$\text{if } i=w, \forall r\in R,(v,w)\in E^{\mathrm{V}}. \quad (4c)$$

Equations (4a), (4b), and (4c) express the flow constraints at ingress, intermediate, and egress substrate routers for $(v,w)\in E^{\mathrm{V}}$, respectively. Equation (4a) means that, if substrate router $i\in N^{\mathrm{S}}$ is the start node for VL of $(v,w)\in E^{\mathrm{V}}$, any of substrate link $(i,j)\in E^{\mathrm{S}}$ is allocated for the VL, i.e., λ_{ij}^{rvw} becomes one. On the contrary, (4c) means that, if router i is the end node for VL of (v,w), any of substrate link $(j,i)\in E^{\mathrm{S}}$ is allocated for the VL, i.e., λ_{ji}^{rvw} becomes one. Equation (4b) represents that the amounts of transmitted and received traffic are equal at router i if i is neither the start nor end node of the VL.

In addition to VLs, access paths are configured on the substrate network. The flow constraints for upstream traffic routed from a DC to an access VR are given by:

$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\theta_{ij,\mathrm{UP}}^{rpv}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\theta_{ji,\mathrm{UP}}^{rpv}=z_{rpv},$$
$$\text{if } a_{rpi}=1 \text{ and } i\neq v,$$
$$\forall r\in R, p\in D_r, i\in N^{\mathrm{S}}, v\in N^{\mathrm{V}}\backslash\{i\}, \quad (5a)$$
$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\theta_{ij,\mathrm{UP}}^{rpv}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\theta_{ji,\mathrm{UP}}^{rpv}=0,$$
$$\text{if } a_{rpi}=1 \text{ and } i=v, \forall r\in R, p\in D_r, i\in N^{\mathrm{S}}, \quad (5b)$$
$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\theta_{ij,\mathrm{UP}}^{rpv}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\theta_{ji,\mathrm{UP}}^{rpv}=0,$$
$$\text{if } a_{rpi}=0 \text{ and } i\neq v,$$
$$\forall r\in R, p\in D_r, i\in N^{\mathrm{S}}, v\in N^{\mathrm{V}}\backslash\{i\}, \quad (5c)$$
$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\theta_{ij,\mathrm{UP}}^{rpv}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\theta_{ji,\mathrm{UP}}^{rpv}=-z_{rpv},$$
$$\text{if } a_{rpi}=0 \text{ and } i=v, \forall r\in R, p\in D_r, i\in N^{\mathrm{S}}. \quad (5d)$$

Equations (5a), (5c), and (5d) express the flow constraints at source, intermediate, and destination substrate routers for an upstream access paths, respectively. Equation (5a) means that, if DC $p\in D_r$ is attached to substrate router $i\in N^{\mathrm{S}}$ and an access VR is not placed on i, an upstream access path departs from i. On the contrary, (5d) means that, if DC p is not attached to router i and an access VR is placed on i, an upstream access path terminates at i. Equation (5c) represents that the amounts of transmitted and received traffic are equal at router i if i is neither the start nor end node of an upstream access path. Equation (5b) means that no substrate links are used to establish an upstream access path if an access VR is located on the router where a DC is attached.

Similarly, the flow constraints for downstream traffic routed from an access VR to a DC are given by:

$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\theta_{ij,\mathrm{DW}}^{rwq}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\theta_{ji,\mathrm{DW}}^{rwq}=z_{rqw},$$
$$\text{if } a_{rqi}=0 \text{ and } i=w, \forall r\in R, q\in D_r, i\in N^{\mathrm{S}}, \quad (6a)$$
$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}}\theta_{ij,\mathrm{DW}}^{rwq}-\sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}}\theta_{ji,\mathrm{DW}}^{rwq}=0,$$
$$\text{if } a_{rqi}=0 \text{ and } i\neq w,$$

$$\forall r \in R, q \in D_r, i \in N^{\mathrm{S}}, w \in N^{\mathrm{V}}\backslash\{i\}, \tag{6b}$$

$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}} \theta^{rwq}_{ij,\mathrm{DW}} - \sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}} \theta^{rwq}_{ji,\mathrm{DW}} = 0,$$

$$\text{if } a_{rqi} = 1 \text{ and } i = w, \forall r \in R, q \in D_r, i \in N^{\mathrm{S}}, \tag{6c}$$

$$\sum_{j\in N^{\mathrm{S}}:(i,j)\in E^{\mathrm{S}}} \theta^{rwq}_{ij,\mathrm{DW}} - \sum_{j\in N^{\mathrm{S}}:(j,i)\in E^{\mathrm{S}}} \theta^{rwq}_{ji,\mathrm{DW}} = -z_{rqw},$$

$$\text{if } a_{rqi} = 1 \text{ and } i \neq w,$$

$$\forall r \in R, q \in D_r, i \in N^{\mathrm{S}}, w \in N^{\mathrm{V}}\backslash\{i\}. \tag{6d}$$

Equations (6a), (6b), and (6d) express the flow constraints at source, intermediate, and destination substrate routers for a downstream access path, respectively. Equation (6c) means that no substrate links are used to establish a downstream access path if an access VR is located on the substrate router where a DC is attached.

Equations (5a)–(6d) enforce that customer traffic from each DC passes through at least one VR. The route of upstream access path is determined by (5a)–(5d); the traffic from source DC $p \in D_r$ is routed to a VR, which is placed on router $v \in N^{\mathrm{V}}$ with $z_{rpv} = 1$. Similarly, the route of downstream access path is determined by (6a)–(6d); the traffic from a VR, which is placed on router $w \in N^{\mathrm{V}}$ with $z_{rqw} = 1$, is routed to destination DC $q \in D_r$. This means that the upstream and downstream access paths are connected via at least one VR.

2) Capacity constraints: The constraints for the router capacity, c^{N}_v, and the substrate link capacity, c^{L}_{ij}, are given by:

$$\sum_{r\in R}\sum_{(p,q)\in P_r} d_{rpq}\left(\sum_{w\in N^{\mathrm{V}}:(v,w)\in E^{\mathrm{V}}} \xi^{rpq}_{vw} + \sum_{w\in N^{\mathrm{V}}:(w,v)\in E^{\mathrm{V}}} \xi^{rpq}_{wv} + \sum_{(i,j)\in E^{\mathrm{S}}:j=v} \theta^{rpv}_{ij,\mathrm{UP}} + \sum_{(i,j)\in E^{\mathrm{S}}:i=v} \theta^{rvq}_{ij,\mathrm{DW}}\right) \leq c^{\mathrm{N}}_v, \quad \forall v \in N^{\mathrm{V}}, \tag{7a}$$

$$\sum_{r\in R}\sum_{(p,q)\in P_r} d_{rpq}\left(\sum_{(v,w)\in E^{\mathrm{V}}} \mu^{rpq}_{vwij} + \sum_{v\in N^{\mathrm{V}}}\left(\theta^{rpv}_{ij,\mathrm{UP}} + \theta^{rvq}_{ij,\mathrm{DW}}\right)\right) \leq c^{\mathrm{L}}_{ij}, \forall (i,j) \in E^{\mathrm{S}}. \tag{7b}$$

The left-hand side of (7a) sums up the amounts of all incoming and outgoing traffics, which include those of VLs and access paths, at router $v \in N^{\mathrm{V}}$. The left-hand side of (7b) sums up the amounts of all traffics, which include those of VLs and access paths, that pass through substrate link $(i,j) \in E^{\mathrm{S}}$. Here, $\mu^{rpq}_{vwij} = \lambda^{rvw}_{ij}\xi^{rpq}_{vw}$ is a new decision variable introduced to express (7b) in a linear form.

$$\mu^{rpq}_{vwij} \leq \lambda^{rvw}_{ij}, \quad \forall r \in R, (p,q) \in P_r, (v,w) \in E^{\mathrm{V}}, (i,j) \in E^{\mathrm{S}}, \tag{8a}$$

$$\mu^{rpq}_{vwij} \leq \xi^{rpq}_{vw}, \quad \forall r \in R, (p,q) \in P_r, (v,w) \in E^{\mathrm{V}}, (i,j) \in E^{\mathrm{S}}, \tag{8b}$$

$$\mu^{rpq}_{vwij} \geq \lambda^{rvw}_{ij} + \xi^{rpq}_{vw} - 1, \quad \forall r \in R, (p,q) \in P_r, (v,w) \in E^{\mathrm{V}}, (i,j) \in E^{\mathrm{S}}. \tag{8c}$$

3) Delay constraints: The delay experienced by the traffic of source-destination DC pair $(p,q) \in P_r$ needs to satisfy the maximum allowable delay, κ_{rpq}. We assume that the experienced delay is composed of the sum of delays at VRs and substrate links through which the traffic of (p,q) passes. The delay constraints for (p,q) in $r \in R$ are given by:

$$\sum_{v\in N^{\mathrm{V}}} \beta_v o^{rpq}_v + \sum_{(i,j)\in E^{\mathrm{S}}} \gamma_{ij}\left(\sum_{(v,w)\in E^{V}} \lambda^{rvw}_{ij} + \sum_{v\in N^{\mathrm{V}}}\left(\theta^{rpv}_{ij,\mathrm{UP}} + \theta^{rvq}_{ij,\mathrm{DW}}\right)\right) \leq \kappa_{rpq}, \quad \forall r \in R, (p,q) \in P_r. \tag{9}$$

The first term of (9) represents the sum of delays at VRs located on the traffic route of (p,q). The second term of (9) represents the sum of delays at substrate links, which are allocated to VLs, an upstream access path, and a downstream access path on the traffic route of (p,q).

4) Other constraints: By the definitions of x_{rv}, o^{rpq}_v, y_{rvw}, z_{rpv}, $\theta^{rpv}_{ij,\mathrm{UP}}$, $\theta^{rwq}_{ij,\mathrm{DW}}$, and ξ^{rpq}_{vw}, we have:

$$y_{rvw} \leq x_{rv}, \forall r \in R, (v,w) \in E^{\mathrm{V}}, \tag{10}$$

$$y_{rvw} \leq x_{rw}, \forall r \in R, (v,w) \in E^{\mathrm{V}}, \tag{11}$$

$$z_{rpv} \leq x_{rv}, \forall r \in R, p \in D_r, v \in N^{\mathrm{V}}, \tag{12}$$

$$\sum_{(i,j)\in E^{\mathrm{S}}} \theta^{rpv}_{ij,\mathrm{UP}} \leq |E^{\mathrm{S}}| \times z_{rpv}, \forall r \in R, p \in D_r, v \in N^{\mathrm{V}}, \tag{13}$$

$$\sum_{(i,j)\in E^{\mathrm{S}}} \theta^{rwq}_{ij,\mathrm{DW}} \leq |E^{\mathrm{S}}| \times z_{rqw}, \forall r \in R, q \in D_r, w \in N^{\mathrm{V}}, \tag{14}$$

$$\xi^{rpq}_{vw} \leq y_{rvw}, \forall r \in R, (p,q) \in P_r, (v,w) \in E^{\mathrm{V}}, \tag{15}$$

$$\xi^{rpq}_{vw} \leq o^{rpq}_v, \forall r \in R, (p,q) \in P_r, (v,w) \in E^{\mathrm{V}}, \tag{16}$$

$$\xi^{rpq}_{vw} \leq o^{rpq}_w, \forall r \in R, (p,q) \in P_r, (v,w) \in E^{\mathrm{V}}, \tag{17}$$

$$o^{rpq}_v \leq x_{rv}, \forall r \in R, (p,q) \in P_r, v \in N^{\mathrm{V}}, \tag{18}$$

$$\sum_{v\in N^{\mathrm{V}}} x_{rv} \leq M_r, \forall r \in R. \tag{19}$$

Equations (10) and (11) mean that a VR is placed on the start and end nodes of VL. Equation (12) means that a router serves a VR if there is a DC that accesses to the router. Equations (13) and (14) ensure that one upstream access path and one downstream access path are configured for each DC. Equation (15) means that a VL is set up on logical link $(v,w) \in E^{\mathrm{V}}$ if there is traffic that passes through (v,w). Equations (16)–(18) are used to determine the value of o^{rpq}_v, which is used in the calculation of delay at (9). Equation (19) limits the number of VRs that can be placed for providing a VN request.

Since we assume that each DC accesses one VR, we have:

$$\sum_{v\in N^{\mathrm{V}}} z_{rpv} = 1, \forall r \in R, p \in D_r. \tag{20}$$

E. Integer linear programming (ILP) problem

Based on the terminologies and assumptions, and the analyses of cost and constraints presented in Sections III-B–III-D, we formulate an optimization problem to minimize the total cost for all VNs as an ILP problem, which is given by:

Objective

$$\min \sum_{r\in R}\sum_{v\in N^{\mathrm{V}}}\Bigg(\alpha_v^{\mathrm{VR}}\sum_{s\in S_{\mathrm{VR}}} h_s^{\mathrm{VR}}\sigma_{srv}^{\mathrm{VR}} + \sum_{w\in N^{\mathrm{V}}:(v,w)\in E^{\mathrm{V}}} \alpha_{vw}^{\mathrm{VL}}\sum_{s\in S_{\mathrm{VL}}} h_s^{\mathrm{VL}}\sigma_{srvw}^{\mathrm{VL}} + \sum_{p\in D_r}\alpha_{rpv}^{\mathrm{AC}}\sum_{s\in S_{\mathrm{AC}}} h_s^{\mathrm{AC}}\sigma_{srvp}^{\mathrm{AC}} + \sum_{(p,q)\in P_r} d_{rpq}\sum_{(i,j)\in E^{\mathrm{S}}}\omega_{ij}\left(\theta_{ij,\mathrm{UP}}^{rpv}+\theta_{ij,\mathrm{DW}}^{rvq}\right)\Bigg) \quad (21a)$$

s.t.

$$\sum_{(p,q)\in P_r} d_{rpq}\left(\sum_{\substack{w\in N^{\mathrm{V}}:\\(v,w)\in E^{\mathrm{V}}}}\xi_{vw}^{rpq}+\sum_{\substack{w\in N^{\mathrm{V}}:\\(w,v)\in E^{\mathrm{V}}}}\xi_{wv}^{rpq}\right) \le \sum_{s\in S_{\mathrm{VR}}}\sigma_{srv}^{\mathrm{VR}} l_s^{\mathrm{VR}}, \forall r\in R, v\in N^{\mathrm{V}}, \quad (21b)$$

$$\sum_{(p,q)\in P_r} d_{rpq}\xi_{vw}^{rpq} \le \sum_{s\in S_{\mathrm{VL}}}\sigma_{srvw}^{\mathrm{VL}} l_s^{\mathrm{VL}}, \forall r\in R, (v,w)\in E^{\mathrm{V}}, \quad (21c)$$

$$z_{rpv}\sum_{q\in D_r:(p,q)\in P_r} d_{rpq} \le \sum_{s\in S_{\mathrm{AC}}}\sigma_{srvp}^{\mathrm{AC}} l_s^{\mathrm{AC}}, \forall r\in R, p\in D_r, v\in N^{\mathrm{V}}, \quad (21d)$$

$$\sum_{s\in S_{\mathrm{VR}}}\sigma_{srv}^{\mathrm{VR}} = 1, \forall r\in R, v\in N^{\mathrm{V}}, \quad (21e)$$

$$\sum_{s\in S_{\mathrm{VL}}}\sigma_{srvw}^{\mathrm{VL}} = 1, \forall r\in R, (v,w)\in E^{\mathrm{V}}, \quad (21f)$$

$$\sum_{s\in S_{\mathrm{AC}}}\sigma_{srvp}^{\mathrm{AC}} = 1, \forall r\in R, v\in N^{\mathrm{V}}, p\in D_r, \quad (21g)$$

$$(3)-(20), \quad (21h)$$

$$\sigma_{srv}^{\mathrm{VR}}\in\{0,1\}, \forall s\in S_{\mathrm{VR}}, r\in R, v\in N^{\mathrm{V}}, \quad (21i)$$

$$\sigma_{srvw}^{\mathrm{VL}}\in\{0,1\}, \forall s\in S_{\mathrm{VL}}, r\in R, (v,w)\in E^{\mathrm{V}}, \quad (21j)$$

$$\sigma_{srvp}^{\mathrm{AC}}\in\{0,1\}, \forall s\in S_{\mathrm{AC}}, r\in R, v\in N^{\mathrm{V}}, p\in D_r, \quad (21k)$$

$$x_{rv}\in\{0,1\}, \forall r\in R, v\in N^{\mathrm{V}}, \quad (21l)$$

$$y_{rvw}\in\{0,1\}, \forall r\in R, (v,w)\in E^{\mathrm{V}}, \quad (21m)$$

$$o_v^{rpq}\in\{0,1\}, \forall r\in R, (p,q)\in P_r, v\in N^{\mathrm{V}}, \quad (21n)$$

$$z_{rpv}\in\{0,1\}, \forall r\in R, p\in D_r, v\in N^{\mathrm{V}}, \quad (21o)$$

$$\lambda_{ij}^{rvw}\in\{0,1\}, \forall r\in R, (v,w)\in E^{\mathrm{V}}, (i,j)\in E^{\mathrm{S}}, \quad (21p)$$

$$\theta_{ij,\mathrm{UP}}^{rpv}\in\{0,1\}, \forall r\in R, p\in D_r, v\in N^{\mathrm{V}}, (i,j)\in E^{\mathrm{S}}, \quad (21q)$$

$$\theta_{ij,\mathrm{DW}}^{rwq}\in\{0,1\}, \forall r\in R, w\in N^{\mathrm{V}}, q\in D_r, (i,j)\in E^{\mathrm{S}}, \quad (21r)$$

$$\xi_{vw}^{rpq}\in\{0,1\}, \forall r\in R, (v,w)\in E^{\mathrm{V}}, (p,q)\in P_r, \quad (21s)$$

$$\mu_{vwij}^{rpq}\in\{0,1\}, \forall r\in R, (p,q)\in P_r, (v,w)\in E^{\mathrm{V}}, (i,j)\in E^{\mathrm{S}}. \quad (21t)$$

$\sigma_{srv}^{\mathrm{VR}}$ denotes a binary decision variable that is set to one if a VR is served on $v\in N^{\mathrm{V}}$ for VN $r\in R$ and imposes cost of class $s\in S_{\mathrm{VR}}$, and zero otherwise. $\sigma_{srvw}^{\mathrm{VL}}$ denotes a binary decision variable that is set to one if VL $(v,w)\in E^{\mathrm{V}}$ is used by VN $r\in R$ and imposes cost of class $s\in S_{\mathrm{VL}}$, and zero otherwise. $\sigma_{srvp}^{\mathrm{AC}}$ denotes a binary decision variable that is set to one if $v\in N^{\mathrm{V}}$ is used as an access VR of VN $r\in R$ and imposes cost of class $s\in S_{\mathrm{AC}}$, and zero otherwise.

If the values of $|D_r|$ and $|P_r|$ do not change with $r\in R$, the number of constraints in the ILP problem is $|R|(|D_r|(2|N^{\mathrm{V}}||N^{\mathrm{S}}|+5|N^{\mathrm{V}}|+1)+|P_r|(3|E^{\mathrm{V}}||E^{\mathrm{S}}|+3|E^{\mathrm{V}}|+2|N^{\mathrm{V}}|+1)+4|E^{\mathrm{V}}|+2|N^{\mathrm{V}}|+|E^{\mathrm{V}}||N^{\mathrm{S}}|+1)+|N^{\mathrm{V}}|+|E^{\mathrm{S}}|$. The number of decision variables in the ILP problem is $|R|(|D_r|(|N^{\mathrm{V}}|(2|E^{\mathrm{S}}|+|S_{\mathrm{AC}}|+1))+|P_r|(|E^{\mathrm{V}}||E^{\mathrm{S}}|+|N^{\mathrm{V}}|+|E^{\mathrm{V}}|)+|N^{\mathrm{V}}|(|S_{\mathrm{VR}}|+1)+|E^{\mathrm{V}}|(|E^{\mathrm{S}}|+|S_{\mathrm{VL}}|+1))$.

The VNDE model contains restrictions which also appear in a typical VNE problem. The feature of bin-packing problem appears in the form of placement problem of VRs. The bin-packing problem is a problem to pack a set of items, each of which has different size, to a number of bins with a fixed capacity. In the case of VNDE model, VRs, each of which handles a certain amount of traffic, are assigned to substrate routers under capacity constraint (7a). The VNDE model aims to suppress the utilization cost of VRs; this limits the use of substrate routers as much as possible. The feature of unsplittable flow problem appears in the routing of traffic flow for each source-destination DC pair in VN requests. A single-flow path, which is composed of an upstream access path, VLs, and a downstream access path, is determined for each source-destination DC pair by the flow constraints (3)–(6d) under capacity constraint (7b).

F. NP-completeness

We prove that the decision version of the VNDE problem (called *VNDE-D* hereinafter) is NP-complete. The VNDE-D problem is defined as follows:

Definition: Given a set of VN requests, R, a substrate network, $G^{\mathrm{S}}(N^{\mathrm{S}},E^{\mathrm{S}})$, logical connections, $G^{\mathrm{V}}(N^{\mathrm{V}},E^{\mathrm{V}})$, the capacity of each substrate link and router, a set of DCs, $D_r, \forall r\in R$, a set of source-destination DC pairs, $P_r, \forall r\in R$, with traffic demands, and cost functions with coefficient parameters, can we find the design and embedding of each VN request in R that makes the total cost be at most C?

Theorem: The VNDE-D problem is NP-complete.

Proof: The VNDE-D problem is in NP, as we can verify whether the total cost is at most C in polynomial time. If an instance of the VNDE-D problem is given, the total cost can be calculated by (21a). Let $D_{\max} = \max_{r\in R} D_r$ and $P_{\max} = \max_{r\in R} P_r$. The total cost can be calculated in $O\left(|R||N^{\mathrm{V}}|\left(|S_{\mathrm{VR}}|+|E^{\mathrm{V}}||S_{\mathrm{VL}}|+D_{\max}|S_{\mathrm{AC}}|+P_{\max}|E^{\mathrm{S}}|\right)\right)$.

We show that the bin-packing problem, which is a known NP-complete problem [53], is reducible to the VNDE-D problem in polynomial time. The bin-packing problem is defined as: given a set of items, U, a size of each item, $s_u, \forall u\in U$, the capacity of bin, B, and a positive integer, Q, is there a partition of U into disjoint sets, $U_1, U_2, \cdots, U_Q$, such that the sum of the item sizes in U_i, $i\in[1,Q]$, is B or less?

First, we construct an instance of the VNDE-D problem from any instance of the bin-packing problem. An instance of the bin-packing problem consists of a set of positive numbers

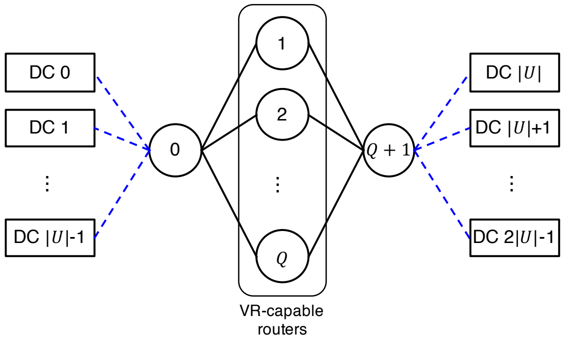


Fig. 3. Substrate network considered in instance construction.

which represent the item sizes, i.e., $\{s_u : u \in [0, |U| - 1]\}$. An instance of the VNDE-D problem is constructed with the following steps.

1) Consider substrate network $G^{\mathrm{S}}(N^{\mathrm{S}}, E^{\mathrm{S}})$ shown in Fig. 3. G^{S} has $Q + 2$ substrate routers. Router 0 is connected to routers $1, 2, \cdots, Q$ by substrate links. Router $Q + 1$ is connected to routers $1, 2, \cdots, Q$ by substrate links. Routers $1, 2, \cdots, Q$ are capable of hosting a VR. Logical connections, G^{V}, is a disconnected graph, where $N^{\mathrm{V}} = \{1, 2, \cdots, Q\}$ and $E^{\mathrm{V}} = \emptyset$.
2) Consider a single VN request, i.e., $|R| = 1$. The VN request has $2|U|$ DCs; DCs $0, 1, \cdots, |U| - 1$ are attached to router 0, and DCs $|U|, |U| + 1, \cdots, 2|U| - 1$ are attached to router $Q + 1$. DCs $0, 1, \cdots, |U| - 1$ generate traffic destined to DCs $|U|, |U|+1, \cdots, 2|U|-1$, respectively, where the traffic amount sent from DC u to DC $u+|U|$ is set to s_u. DCs $|U|, |U|+1, \cdots, 2|U|-1$ generate nominal traffic destined to DCs $0, 1, \cdots, |U|-1$, respectively, where the traffic amount is set to zero. In summary, there are $2|U|$ traffic flows in the VN; $P_r = \{(0, |U|), (1, |U| + 1), \cdots, (|U| - 1, 2|U| - 1), (|U|, 0), (|U| + 1, 1), \cdots, (2|U| - 1, |U| - 1)\}$.
3) The capacity of routers $1, 2, \cdots, Q$ is set to $2B$. The capacity of routers 0 and $Q+1$ is set to $\sum_{u\in[0,|U|-1]} s_u$. The capacity of each substrate link is set to $\sum_{u\in[0,|U|-1]} s_u$.
4) Set two cost classes for $f_{\mathrm{AC}}(t)$ as follows:

$$f_{\mathrm{AC}}(t) = \begin{cases} 0, & \text{if } t = 0, \\ \frac{C}{|U|}, & \text{otherwise.} \end{cases} \quad (22)$$

$f_{\mathrm{VR}}(t)$ and $f_{\mathrm{VL}}(t)$ are set to zero for all possible values of traffic t. $\omega_{ij} = 0$ for all substrate links in E^{S} so that the access path cost is zero. Set $\alpha^{\mathrm{AC}}_{rpv} = 1$ for all DC $p \in D_r$ and router $v \in N^{\mathrm{V}}$.

The above steps transform any instance of the bin-packing problem into an instance of the VNDE-D problem in a polynomial complexity of $O(|U|Q)$.

If an instance of the bin-packing problem is a Yes instance, there is a partition of U into $U_1, U_2, \cdots, U_Q$ such that the sum of the item sizes in U_i, $i \in [1, Q]$, is B or less. In the corresponding VNDE-D instance, the traffic flow departing from DC $u \in [0, |U|-1]$ is routed so that a VR placed on router $i \in [1, Q]$ becomes the access VR of DC u. For each traffic flow departing from DC $u \in [0, |U|-1]$, the corresponding VR receives incoming traffic of s_u units and transmits outgoing traffic of s_u units; the VR processes $2s_u$ traffic of DC u in total. The total amount of traffic that each VR processes is less than or equal to $2B$, i.e., the router capacity. The cost of using access VR corresponding to DC $u \in [0, |U|-1]$ is $\frac{C}{|U|}$; the total cost is less than or equal to C. Traffic flow departing from DC $u \in [|U|, 2|U| - 1]$, whose traffic amount is zero, is routed so that any one of VRs becomes an access VR; it does not make any impact on the router capacity nor the cost. Therefore, there is a design and embedding of the VN request that makes the total cost be at most C; the corresponding VNDE-D instance is a Yes instance.

Conversely, if an instance of the VNDE-D problem is a Yes instance, there is a design and embedding of the VN request that makes the total cost be at most C. According to node capacity constraint (7a), the amount of traffic that a VR placed on each router processes is at most $2B$. Since VLs cannot be established between VRs in the network shown in Fig. 3, each traffic flow must be sent to one of the VRs by using an upstream access path and transferred to the destination DC by using a downstream access path. For each traffic flow departing from DC $u \in [0, |U| - 1]$, the amounts of incoming traffic and outgoing traffic at the corresponding VR are s_u and s_u, respectively. This means that, the traffic departing from DCs $0, 1, \cdots, |U| - 1$ are assigned to VRs in routers $1, 2, \cdots, Q$ so that the total incoming traffic at each router is at most B. Therefore, the corresponding bin-packing problem instance is a Yes instance.

This confirms that the bin-packing problem, which is NP-complete, is polynomial time reducible to the VNDE-D problem. Since the VNDE-D problem is in NP, the VNDE-D problem is NP-complete. □

IV. Heuristic algorithms

Generally, a network operator needs to obtain suitable VN graphs and their embedding within a certain time period so that the operator can provide VNs in response to customers' needs. When the problem size becomes large, i.e., the number of requests or the network size increases, the computation time required for solving an ILP problem tends to increase. The ILP problem is intractable in the case that the computation time exceeds the time period available to the operator. For instance, in an experiment with a 15-node substrate network and three VN requests, each of which has five DCs, the ILP problem requires more than 17 hours to be solved when using a server with an Intel Xeon Silver 4114 processor with 192 GB memory.

Our solution is to introduce two heuristic algorithms that determine the VNDE in a greedy manner.

The first algorithm determines the VNDE for each VN request one by one. We call the first algorithm *Greedy-A* hereinafter. Greedy-A uses a modified version of the ILP problem in Section III-E, in which the number of VN requests, $|R|$, is set to one and the number of VRs is fixed to a certain value. Let M^*_r denote the number of VRs placed for VN request $r \in R$. The modified ILP problem replaces (19) with $\sum_{v\in N^{\mathrm{V}}} x_{rv} = M^*_r, \forall r \in R$. The algorithm examines possible

Algorithm 1: Greedy-A

for *VN request* $r \in R$ *in descending order of total traffic demand* **do**
- $M_r^* \longleftarrow 1$
- **while** $M_r^* \leq M_r$ **do**
  - Solve the ILP problem of (21a)–(21t) by fixing the number of VRs to M_r^* and setting $|R| = 1$.
  - Record the VNDE with its cost.
  - Increment M_r^* by one.
- Select the number of VRs that achieves the smallest cost for VN request r.
- Configure a VN and access paths for r.
- Set the capacity of each substrate link, c_{ij}^{L}, and that of each router, c_v^{N}, to the remaining capacities of them.

Return the total cost required to configure all VNs.

Algorithm 2: Greedy-B

for *VN request* $r \in R$ *in descending order of total traffic demand* **do**
- $M_r^* \longleftarrow 1$
- **while** $M_r^* \leq M_r$ **do**
  - Determine the placement of VRs for VN request r, x_{rv}, by solving the ILP problem of (23a)–(23c).
  - Solve the ILP problem of (21a)–(21t) by fixing the number of VRs to M_r^*, setting $|R| = 1$, and assuming that x_{rv} is a given parameter.
  - Record the VNDE with its cost.
  - Increment M_r^* by one.
- Select the number of VRs that achieves the smallest cost for VN request r.
- Configure a VN and access paths for r.
- Set the capacity of each substrate link, c_{ij}^{L}, and that of each router, c_v^{N}, to the remaining capacities of them.

Return the total cost required to configure all VNs.

numbers of VRs for each VN request, i.e., it solves the modified ILP problem by changing M_r^* from one to M_r. Once the VNDE is determined for a VN request, it is never changed. The procedure of Greedy-A is described in Algorithm 1. If the number of VN requests is so small that the usages of substrate links and routers are not tight, Greedy-A outputs the VNDE with the minimum total cost.

Greedy-A solves the modified ILP problem $\sum_{r \in R} M_r$ times. The modified ILP problem has $|D_r|(2|N^{\mathrm{V}}||N^{\mathrm{S}}| + 5|N^{\mathrm{V}}| + 1) + |P_r|(3|E^{\mathrm{V}}||E^{\mathrm{S}}| + 3|E^{\mathrm{V}}| + 2|N^{\mathrm{V}}| + 1) + 4|E^{\mathrm{V}}| + 2|N^{\mathrm{V}}| + |E^{\mathrm{V}}||N^{\mathrm{S}}| + 1 + |N^{\mathrm{V}}| + |E^{\mathrm{S}}|$ constraints and $|D_r|(|N^{\mathrm{V}}|(2|E^{\mathrm{S}}| + |S_{\mathrm{AC}}| + 1)) + |P_r|(|E^{\mathrm{V}}||E^{\mathrm{S}}| + |N^{\mathrm{V}}| + |E^{\mathrm{V}}|) + |N^{\mathrm{V}}|(|S_{\mathrm{VR}}| + 1) + |E^{\mathrm{V}}|(|E^{\mathrm{S}}| + |S_{\mathrm{VL}}| + 1)$ decision variables, which are smaller than those of the ILP problem shown in Section III-E.

The second algorithm aims to further reduce the computation time compared to Greedy-A by separately determining the placement of VRs. We call the second algorithm *Greedy-B* hereinafter. For each possible number of VRs, Greedy-B first determines the placement of VRs, x_{rv}, based on the length of shortest paths between each DC and the corresponding access VR; we expect that access paths will be configured along these shortest paths. The algorithm then solves the modified ILP problem by assuming that x_{rv} is given. Let $d(i, v)$ denote the length of shortest path from $i \in N^{\mathrm{S}}$ to $v \in N^{\mathrm{V}}$. The VR placement problem for VN request r is given by:

Objective

$$\min \sum_{p \in D_r} \sum_{i \in N^{\mathrm{S}}} \sum_{v \in N^{\mathrm{V}}} d(i, v) a_{rpi} x_{rv}, \tag{23a}$$

s.t.

$$\sum_{v \in N^{\mathrm{V}}} x_{rv} = M_r^*, \tag{23b}$$

$$x_{rv} \in \{0, 1\}, \forall v \in N^{\mathrm{V}}. \tag{23c}$$

The objective function in (23a) represents the sum of shortest path lengths between a substrate router where a DC is attached and its corresponding access VR. The procedure of Greedy-B is described in Algorithm 2.

The VR placement problem of (23a)–(23c) has one constraints and $|N^{\mathrm{V}}|$ decision variables. By determining the placement of VRs in advance, the size of the modified ILP problem is reduced compared to that of Greedy-A; it has $|D_r|(|N^{\mathrm{V}}|(2|E^{\mathrm{S}}| + |S_{\mathrm{AC}}| + 1)) + |P_r|(|E^{\mathrm{V}}||E^{\mathrm{S}}| + |N^{\mathrm{V}}| + |E^{\mathrm{V}}|) + |N^{\mathrm{V}}||S_{\mathrm{VR}}| + |E^{\mathrm{V}}|(|E^{\mathrm{S}}| + |S_{\mathrm{VL}}| + 1)$ decision variables.

V. Benchmark algorithm

For the purpose of evaluating the performance of the VNDE model, we introduce a benchmark algorithm. We design the benchmark algorithm based on a classic VNE approach [12], [13]. Since the VNE approach only determines the embedding of given network graphs, the VN graph design needs to be determined in advance; the benchmark algorithm solves (1) VN graph design problem and (2) VNE problem separately. The first step determines a network graph for each VN request in R. The second step embeds the network graphs constructed in the first step into the substrate network. We determine the embedding of VN graphs by solving an ILP problem rather than using algorithms in recent VNE works. This is because we aim to compare the performance of provisioning cost without being affected by features of each particular VNE algorithm. In addition, the VNE algorithms presented in existing literatures cannot be adopted to the VN embedding in the VNDE problem, which needs to determine the assignment of VRs to DCs and the routing of access paths in addition to the mapping of VRs and VLs.

A. VN graph design problem

We set the number of VRs for VN request $r \in R$, M_r^*, in advance. If M_r^* is set to more than one, we determine the connections between VRs with the expectation that the utilization cost of transit VRs and VLs will be suppressed in the second step. Since the assignment of VRs to DCs has not been determined at this step, we need to design VN graphs based on an assumption of traffic from DCs to VRs. If the assignment determined in the second step differs from that assumed in this step, the access path cost can become large. In this benchmark algorithm, we design VN graphs by simply assuming that the sum of traffic of DCs is evenly loaded on each VR. The amount of traffic loaded on each VR is denoted by d^{AVG}.

We consider directed graph $G^{\mathrm{D}}(N^{\mathrm{D}}, E^{\mathrm{D}})$ to determine a network graph for each VN request. N^{D} is a set of VRs, where $|N^{\mathrm{D}}| = M_r^*$. E^{D} consists of all possible connections between VRs in N^{D}, i.e., G^{D} is a full-mesh network. Each VR in N^{D} generates traffic of $d^{\mathrm{AVG}} = \frac{\sum_{(p,q) \in P_r} d_{rpq}}{M_r^*(M_r^*-1)}$ to each of other VRs. We introduce decision variable $t_{vw}^{v'w'}$ to represent the traffic amount from VR $v' \in N^{\rm D}$ to VR $w' \in N^{\rm D}$ that passes through $(v, w) \in E^{\rm D}$.

For each VN request in R, we obtain the traffic amount that passes through each link in $E^{\rm D}$ by solving the following optimization problem.

$$\text{Objective}$$
$$\min \sum_{(v,w)\in E^{\rm D}} \alpha_{vw}^{\rm VL} \sum_{s\in S_{\rm VL}} h_s^{\rm VL}\sigma_{svw}^{\rm VL} \tag{24a}$$
$$\text{s.t.}$$
$$\sum_{v'\in N^{\rm D}} \sum_{w'\in N^{\rm D}\backslash\{v'\}} t_{vw}^{v'w'} \leq \sum_{s\in S_{\rm VL}} \sigma_{svw}^{\rm VL} l_s^{\rm VL}, \forall (v,w)\in E^{\rm D}, \tag{24b}$$
$$\sum_{s\in S_{\rm VL}} \sigma_{svw}^{\rm VL} = 1, \forall (v,w)\in E^{\rm D}, \tag{24c}$$
$$\sum_{\substack{w\in N^{\rm D}:\\(v,w)\in E^{\rm D}}} t_{vw}^{v'w'} - \sum_{\substack{w\in N^{\rm D}:\\(w,v)\in E^{\rm D}}} t_{wv}^{v'w'} = \begin{cases} d^{\rm AVG} & \text{if } v=v',\\ -d^{\rm AVG} & \text{if } v=w',\\ 0 & \text{otherwise,}\end{cases}$$
$$\forall v'\in N^{\rm D}, w'\in N^{\rm D}\backslash\{v'\}, \tag{24d}$$
$$\sum_{v'\in N^{\rm D}} \sum_{w'\in N^{\rm D}\backslash\{v'\}} \left(\sum_{\substack{w\in N^{\rm D}:\\(v,w)\in E^{\rm D}}} t_{vw}^{v'w'} + \sum_{\substack{w\in N^{\rm D}:\\(w,v)\in E^{\rm D}}} t_{wv}^{v'w'} \right) \leq c_{\rm MAX}^{\rm N},$$
$$\forall v\in N^{\rm D}, \tag{24e}$$
$$\sum_{v'\in N^{\rm D}} \sum_{w'\in N^{\rm D}\backslash\{v'\}} t_{vw}^{v'w'} \leq c_{\rm MAX}^{\rm L}, \forall (v,w)\in E^{\rm D}, \tag{24f}$$
$$\sigma_{svw}^{\rm VL} \in \{0,1\}, \forall s\in S_{\rm VL}, (v,w)\in E^{\rm D}, \tag{24g}$$
$$t_{vw}^{v'w'} \geq 0, \forall v'\in N^{\rm D}, w'\in N^{\rm D}\backslash\{v'\}, (v,w)\in E^{\rm D}. \tag{24h}$$

Objective function (24a) attempts to minimize the sum of utilization cost of VLs. Note that the sum of utilization cost of transit VRs, namely $\sigma_{sv}^{\rm VR}$, does not appear in (24a) since it is taken to be a constant value in this problem. Equations (24b) and (24c) determine the cost class of VL $(v, w) \in E^{\rm D}$. Equation (24d) is a flow constraint for each traffic transferred between VRs. Equation (24e) is the capacity constraint for each VR, where $c_{\rm MAX}^{\rm N}$ is a given parameter that represents the largest capacity among VRs in the network. Equation (24f) is the capacity constraint for each VL, where $c_{\rm MAX}^{\rm L}$ is a given parameter that represents the largest capacity among VLs in the network. Equations (24g) and (24h) are definitions of decision variables.

A network graph for VN request $r \in R$ is constructed by connecting VRs in $N^{\rm D}$ by using links in $E^{\rm D}$ that satisfy $\sum_{v'\in N^{\rm D}} \sum_{w'\in N^{\rm D}\backslash\{v'\}} t_{vw}^{v'w'} > 0$. The constructed network graphs, $G_r^{\rm VN}(N_r^{\rm VN}, E_r^{\rm VN}), \forall r \in R$, become inputs of the VNE problem in the next step.

B. VNE problem

The goal of this step is to determine the embedding of all network graphs constructed in the previous step, $G_r^{\rm VN}(N_r^{\rm VN}, E_r^{\rm VN}), \forall r \in R$, into the substrate network. We determine the assignment and the traffic route at the same time with the VNE in this problem.

We newly introduce the following decision variables. δ_{rfv} is a binary decision variable that is set to one if VR $f \in N_r^{\rm VN}$ of VN request $r \in R$ is mapped to router $v \in N^{\rm V}$, and zero otherwise. ϵ_{rfgvw} is a binary decision variable that is set to one if link $(f, g) \in E_r^{\rm VN}$ of VN request $r \in R$ is mapped to logical link $(v, w) \in E^{\rm V}$, and zero otherwise.

We obtain the VNE by solving the following optimization problem.

$$\text{Objective}$$
$$\min (21\text{a}) \tag{25a}$$
$$\text{s.t.}$$
$$(21\text{b})-(21\text{g}), (3)-(18), \tag{25b}$$
$$\sum_{v\in N^{\rm V}} x_{rv} = M_r^*, \forall r\in R, \tag{25c}$$
$$(20), (21\text{i})-(21\text{s}), \tag{25d}$$
$$\delta_{rfv}\delta_{rgw} = \epsilon_{rfgvw},$$
$$\forall r\in R, f\in N_r^{\rm VN}, g\in N_r^{\rm VN} : (f,g)\in E_r^{\rm VN},$$
$$v\in N^{\rm V}, w\in N^{\rm V} : (v,w)\in E^{\rm V}, \tag{25e}$$
$$\sum_{f\in N_r^{\rm VN}} \delta_{rfv} = x_{rv}, \forall r\in R, v\in N^{\rm V}, \tag{25f}$$
$$\sum_{v\in N^{\rm V}} \delta_{rfv} = 1, \forall r\in R, f\in N_r^{\rm VN}, \tag{25g}$$
$$\delta_{rfv} \leq \sum_{p\in D_r} z_{rpv}, \forall r\in R, f\in N_r^{\rm VN}, v\in N^{\rm V}, \tag{25h}$$
$$\epsilon_{rfgvw} \leq y_{rvw}, \forall r\in R, (f,g)\in E_r^{\rm VN}, (v,w)\in E^{\rm V}, \tag{25i}$$
$$\delta_{rfv} \in \{0,1\}, \forall r\in R, f\in N_r^{\rm VN}, v\in N^{\rm V}, \tag{25j}$$
$$\epsilon_{rfgvw} \in \{0,1\}, \forall r\in R, (f,g)\in E_r^{\rm VN}, (v,w)\in E^{\rm V}. \tag{25k}$$

Equation (25c) corresponds to (19), but ensures that the number of placed VRs is exactly M_r^*. Equation (25e) ensures that link $(f, g) \in E_r^{\rm VN}$ is mapped to logical link $(v, w) \in E^{\rm V}$ if VRs $f \in N_r^{\rm VN}$ and $g \in N_r^{\rm VN}$ are mapped to routers $v \in N^{\rm V}$ and $w \in N^{\rm V}$, respectively. Equation (25e) can be linearized as follows:

$$\epsilon_{rfgvw} \leq \delta_{rfv},$$
$$\forall r\in R, f\in N_r^{\rm VN}, g\in N_r^{\rm VN} : (f,g)\in E_r^{\rm VN},$$
$$v\in N^{\rm V}, w\in N^{\rm V} : (v,w)\in E^{\rm V}, \tag{26a}$$
$$\epsilon_{rfgvw} \leq \delta_{rgw},$$
$$\forall r\in R, f\in N_r^{\rm VN}, g\in N_r^{\rm VN} : (f,g)\in E_r^{\rm VN},$$
$$v\in N^{\rm V}, w\in N^{\rm V} : (v,w)\in E^{\rm V}, \tag{26b}$$
$$\epsilon_{rfgvw} \geq \delta_{rfv} + \delta_{rgw} - 1,$$
$$\forall r\in R, f\in N_r^{\rm VN}, g\in N_r^{\rm VN} : (f,g)\in E_r^{\rm VN},$$
$$v\in N^{\rm V}, w\in N^{\rm V} : (v,w)\in E^{\rm V}. \tag{26c}$$

Equations (25f) and (25g) state the mapping of VRs. Equation (25h) ensures that at least one DC is assigned to each VR. Equation (25i) specifies the mapping of VLs. Equations (25j) and (25k) are definitions of decision variables.

VI. Numerical results

This section presents numerical results of VN provisioning performed by using the proposed VNDE model. First, we

TABLE V
COST CLASSES FOR $f_{\mathrm{VR}}(t)$.

Class	0	1	2	3
Traffic amount [Gbps]	0	(0, 100]	(100, 500]	(500, 1000]
Cost	0	10	20	30

TABLE VI
COST CLASSES FOR $f_{\mathrm{VL}}(t)$.

Class	0	1	2	3
Traffic amount [Gbps]	0	(0, 10]	(10, 50]	(50, 100]
Cost	0	10	20	30

TABLE VII
COST CLASSES FOR $f_{\mathrm{AC}}(t)$.

Class	0	1	2	3
Traffic amount [Gbps]	0	(0, 100]	(100, 500]	(500, 1000]
Cost	0	10	20	30

evaluate the performance of VNDE model compared with the benchmark model by applying both to a small-size network in Section VI-A. After that, we evaluate the performance of VNDE model as applied to SINET5 in Section VI-B.

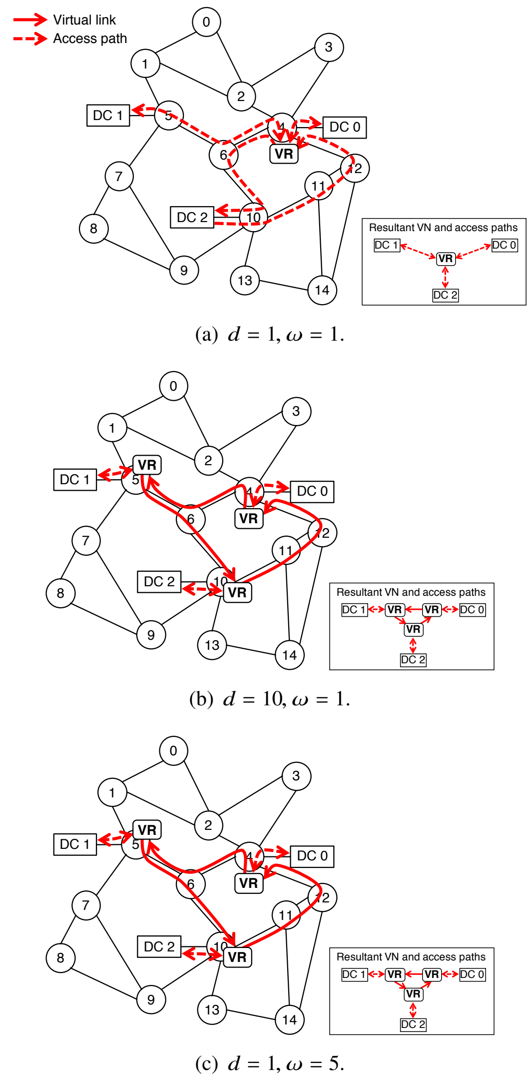
A. Small-size network

1) Simulation environment: The Atlanta network [54], which has 15 nodes and 44 directed links, is used as a substrate network. The capacity of each substrate link, c^{L}_{ij}, is set to 100 Gbps. We assume that all substrate routers can be VRs, i.e., $N^{\mathrm{V}} = N^{\mathrm{S}}$. G^{V} is considered as a full-meshed network. The capacity of each router, c^{N}_{v}, is set to 1000 Gbps. We set four cost classes for transit VRs, VLs, and access VRs. The utilization costs of transit VR, VL, and access VR are set as shown in Tables V, VI, and VII, respectively. We set $\alpha^{\mathrm{VR}}_{v} = 1$, $\alpha^{\mathrm{VL}}_{vw} = 1$, and $\alpha^{\mathrm{AC}}_{rpv} = 1$. The maximum allowable number of VRs in each VN, M_r, is set to the number of DCs that belong to VN request $r \in R$. The maximum allowable delay of each source-destination DC pair, κ_{rpq}, is set to 10 time units. The delay at each VR, β_v, is set to 1 time unit. The delay of each substrate link, γ_{ij}, is set to 1 time unit.

We assume that every DC pair requests the same traffic amount, i.e., $d_{rpq} = d$ Gbps, $\forall r \in R, (p, q) \in P_r$. All substrate links are assumed to have the same utilization cost per unit transmission capacity, i.e., $\omega_{ij} = \omega$ cost unit/Gbps, $\forall (i, j) \in E^{\mathrm{S}}$. In the proposed model, we obtain the VN provisioning by solving the ILP problem and running Greedy-A and Greedy-B. In the benchmark model, the number of VRs for VN request $r \in R$, M^{*}_{r}, is set to one, two, and three. We set the upper limit of computation time to 1.0×10^{5} s in CPU time.

Simulations in this section use a server with an Intel Xeon Silver 4114 processor with 192 GB memory. ILP problems are solved by CPLEX 20.1.0.0 [55].

2) Examples of VN graph design and embedding: Fig. 4 shows examples of resultant VNs obtained by solving the ILP problem of the VNDE model when the number of VNs, $|R|$, is set to one. In these examples, customer's DCs are connected to substrate routers 4, 5, and 10. Solid arrow lines denote VLs, which compose a VN of the customer. Dashed arrow lines denote access paths, which connect DCs to VRs. In Fig. 4(a), where the traffic demand, d, and the utilization cost of substrate link per unit transmission capacity, ω, are set to 1 Gbps and 1 cost unit/Gbps, respectively, only one VR is placed on the network and all DCs are connected to the VR. In Fig. 4(b), where $d = 10$ Gbps and $\omega = 1$ cost unit/Gbps, VRs are placed on the routers that have DCs. In the later case, data traffic destined to the same DC is once aggregated at a VR and then transferred to the DC through VLs. The same trend can be observed when ω is set to a larger value, as shown in Fig. 4(c). These examples indicate that the proposed model designs VN graphs and embeds them into the substrate infrastructure according to the volume of traffic demands and access path cost. Note that, in Fig. 4(a), different access routes are allocated to upstream and downstream traffic between DC 2 and the VR; we allow such an allocation to improve the flexibility in infrastructure resource utilization.



(a) $d = 1, \omega = 1$.

(b) $d = 10, \omega = 1$.

(c) $d = 1, \omega = 5$.

Fig. 4. Examples of resultant VNDE for Atlanta network ($|D_r| = 3$ and $|R| = 1$).

3) Comparison between VNDE and benchmark models: Next, we present the performance of proposed VNDE model, compared with that of the benchmark model, by varying

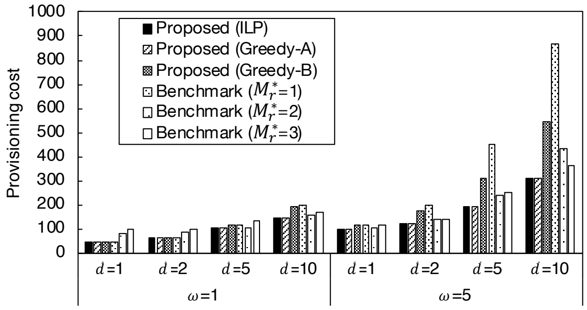


Fig. 5. Average provisioning cost ($|D_r| = 3$ and $|R| = 1$).

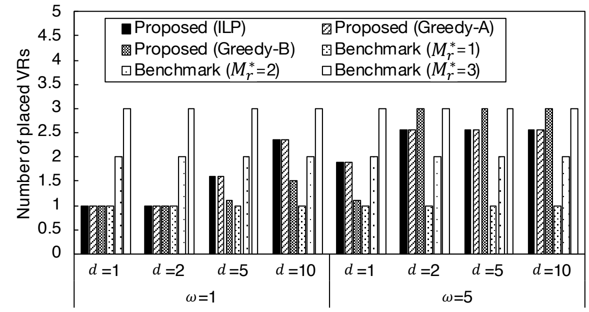


Fig. 6. Average number of placed VRs ($|D_r| = 3$ and $|R| = 1$).

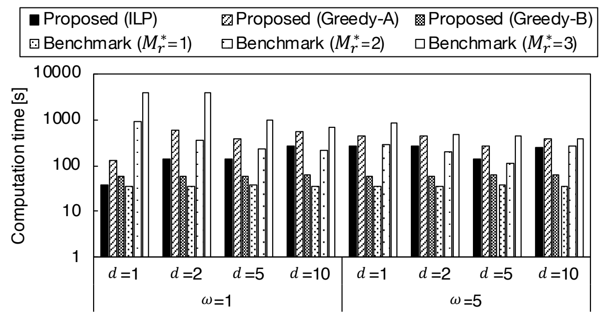


Fig. 7. Average computation time ($|D_r| = 3$ and $|R| = 1$).

parameters, such as d, ω, $|D_r|$, and $|R|$. For each VN request, a substrate router to which each DC is connected is randomly selected from among all substrate routers such that the selected routers do not overlap among the DCs. We perform 20 trials for each parameter setting, and obtain averages of provisioning cost, number of placed VRs, and computation time (CPU time).

Fig. 5 shows the average provisioning cost when the number of DCs is set to $|D_r| = 3$ and the number of VN requests is set to $|R| = 1$. In both proposed and benchmark models, the provisioning cost increases as traffic demand d increases. This is because, as d increases, higher cost classes tend to be used in the deployment of VRs and VLs. In the proposed model which solves the ILP problem, the provisioning cost increases at most 2.1 times, but not 5 times, in $\omega = 5$ cost units/Gbps, compared to that of $\omega = 1$ cost unit/Gbps. This means that the proposed model suppresses the provisioning cost by using multiple VRs and VLs and shortening access paths in the case of $\omega = 5$ cost units/Gbps. In the benchmark model, the value of M_r^* yielding the lowest provisioning cost changes depending on the values of ω and d. For example, $M_r^* = 2$ leads to the lowest provisioning cost among the three when $(\omega, d) = (1, 10)$ whereas $M_r^* = 3$ is the best when $(\omega, d) = (5, 10)$. The proposed model which solves the ILP problem achieves the same or lower provisioning cost compared to the benchmark algorithm with any settings of M_r^*. For example, compared to the benchmark with $M_r^* = 2$, the proposed model reduces the provisioning cost by 19.8% when $(\omega, d) = (5, 5)$. In the scenarios where the proposed model with the ILP approach gets lower cost, i.e., $(\omega, d) = (1, 5)$, $(1, 10)$, $(5, 1)$, $(5, 2)$, $(5, 5)$, and $(5, 10)$, two or more VRs can be placed for the VN request, as shown later in Fig. 6. Greedy-A and Greedy-B achieve lower provisioning cost compared to one or more results of the benchmark algorithm; this means that the network operators can deploy cost-efficient VNs without concerning how to choose appropriate value of M_r^*. Since the number of VN requests, $|R|$, is set to one, Greedy-A outputs the same result as that obtained by solving the ILP problem. The difference between the provisioning costs obtained by Greedy-A and Greedy-B is at most 77%. As shown later, this difference in performance between Greedy-A and Greedy-B becomes small when $|D_r|$ increases.

Fig. 6 shows the average number of placed VRs when $|D_r| = 3$ and $|R| = 1$. As described before, in the benchmark model, the number of VRs is fixed to one, two, or three. From Figs. 5 and 6, it can be seen that the proposed model adequately chooses the number of placed VRs yielding lower provisioning cost than the benchmark model. In the proposed model, only one VR is deployed for a VN when $1 \leq d \leq 2$ Gbps and $\omega = 1$ cost unit/Gbps; all DCs are connected to the same VR to exchange data with each other. The number of VRs can be more than one in some scenarios with larger d. In these scenarios, data of DCs are exchanged via VRs and VLs; the provisioning cost can be suppressed by deploying multiple VRs and setting VLs. When $\omega = 5$ cost units/Gbps, the number of VRs can be more than one when $d = 1$ and 2 Gbps due to the increase in access path cost. Greedy-B also changes the number of placed VRs according to the values of d and ω.

Fig. 7 shows the average computation time to obtain the VN provisioning when $|D_r| = 3$ and $|R| = 1$. Table VIII shows the standard deviations corresponding to the results in Fig. 7. Note that the computation time includes overheads other than the time to solve the ILP problem, such as the time required to generate constraints according to input parameters. Except for the case with $d = 1$ and $\omega = 1$, the benchmark algorithm with $M_r^* = 1$ attains the shortest computation time, followed by the proposed model with Greedy-B. Greedy-B and the benchmark model with $M_r^* = 1$ achieve smaller deviations than the other models. In this evaluation, where $|R| = 1$, Greedy-A takes a longer computation time than solving the ILP problem. The ILP problem solved in Greedy-A is the same with the original ILP problem except that the number of VRs is fixed to a certain value, M_r^*. Since the number of DCs, $|D_r|$, is set to three, Greedy-A solves the ILP problem three times, i.e., $M_r^* = 1$, 2,

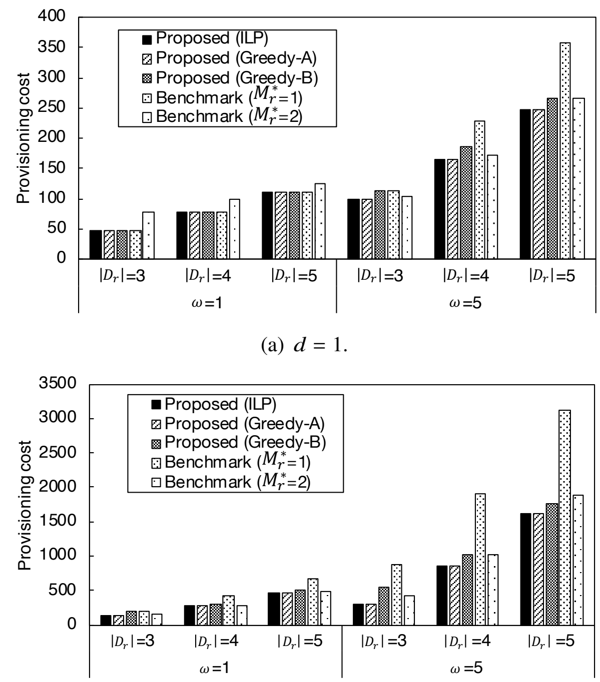
TABLE VIII
STANDARD DEVIATIONS OF COMPUTATION TIME [S] CORRESPONDING TO FIG. 7 ($|D_r| = 3$ AND $|R| = 1$).

	$\omega = 1$				$\omega = 5$			
	$d = 1$	$d = 2$	$d = 5$	$d = 10$	$d = 1$	$d = 2$	$d = 5$	$d = 10$
Proposed (ILP)	5.63	121.89	73.80	181.77	216.56	190.40	126.01	226.39
Proposed (Greedy-A)	20.93	463.30	353.68	388.14	246.30	235.50	167.94	246.71
Proposed (Greedy-B)	0.46	0.39	0.45	0.55	0.42	0.29	0.57	0.87
Benchmark ($M_r^* = 1$)	0.29	0.23	0.31	0.24	0.27	0.26	0.26	0.31
Benchmark ($M_r^* = 2$)	322.95	113.87	373.91	115.50	268.92	183.19	117.23	356.14
Benchmark ($M_r^* = 3$)	1367.65	1186.69	1004.03	785.28	951.05	598.79	386.77	511.57

and 3, and selects the number of VRs that achieves the smallest cost. As shown later, the computation time of the ILP approach exceeds that of Greedy-A when $|R|$ and $|D_r|$ increase. On the other hand, Greedy-B takes a longer computation time than the ILP approach when $d = 1$ and $\omega = 1$. In this parameter setting, where the traffic demand of VN request and the link utilization cost are small enough, the optimization solver (i.e., CPLEX) can find the optimal solution in a short time. Greedy-B solves an ILP problem six times, i.e., the VR placement problem and the modified VNDE problem for $M_r^* = 1$, 2, and 3. This increases the computation time of Greedy-B compared to solving the original ILP problem of the VNDE model when $d = 1$ and $\omega = 1$. The benchmark model with $M_r^* = 2$ and 3 takes longer time to complete the computation than that with $M_r^* = 1$; one of the reasons is that the benchmark model needs to solve the VN graph design problem in (24a)–(24h) when M_r^* is set to more than one.

Figs. 8 and 9 show the average provisioning cost and the average number of placed VRs, respectively, when the number of DCs, $|D_r|$, is set to three, four, or five. We do not show results of the benchmark model with $M_r^* = 3$ since the computation time exceeds 1.0×10^5 s in multiple trials. The traffic amount is set to $d = 1$ in Figs. 8(a) and 9(a), and $d = 10$ in Figs. 8(b) and 9(b). The provisioning cost increases as $|D_r|$ becomes large since the total amount of traffic exchanged between DCs increases. The larger the number of DCs, the greater the merit of setting multiple VRs and aggregating traffic on them. For example, compared to the benchmark with $M_r^* = 2$, the proposed model reduces the provisioning cost by 28.7% when $d = 10$, $\omega = 5$, and $|D_r| = 3$. The difference between the provisioning costs obtained by Greedy-A and Greedy-B tends to become small as $|D_r|$ increases; the difference is at most 20% and 9.5% when $|D_r| = 4$ and 5, respectively. Note that, in some scenarios in Figs. 8 and 9, the results of total provisioning cost become similar among the examined models though the number of VRs is different. As described in Section III-C, the total provisioning cost consists of four types of costs: the costs of transit VRs, VLs, access VRs, and access paths. Some of them can be related to each other; for example, the reduction in the number of VRs leads to the extension of access path length. If the change in the VR cost is almost the same as those in the other costs, the resultant provisioning cost can be similar.

Fig. 10 shows the average computation time to obtain the VN provisioning when $|D_r|$ is set to three, four, or five. Tables IX and X show the standard deviations corresponding to the results in Figs. 10(a) and 10(b), respectively. The computation time increases according to $|D_r|$ in both proposed and benchmark models. Similar to the results in Table VIII, Greedy-B and the benchmark model with $M_r^* = 1$ achieve smaller deviations than the other models.



(a) $d = 1$.

(b) $d = 10$.

Fig. 8. Average provisioning cost depending on number of DCs ($|R| = 1$).

We evaluate the performance of the proposed model and the benchmark model with $M_r^* = 1$ by varying the number of VNs, $|R|$. The number of DCs per VN is set to $|D_r| = 4$ or 5. The traffic amount is set to $d = 1$ or 5. We fix $\omega = 1$ in this evaluation.

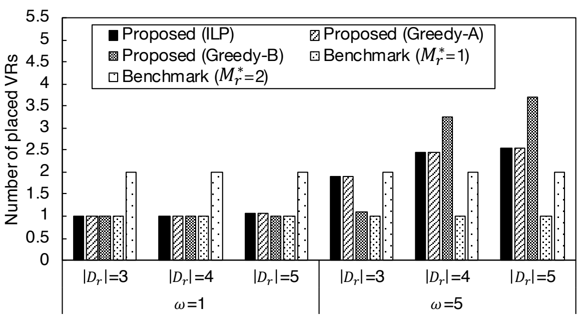
Figs. 11 and 12 show the average provisioning cost per VN and the average number of placed VRs per VN, respectively, when the number of VNs is set to $2 \le |R| \le 5$. We show the results of the proposed model with the ILP approach only in $d = 1$; in $d = 5$, the computation time exceeds 1.0×10^5 s in multiple trials. Fig. 11(a) shows that, in the case of $d = 1$, only one VR is deployed for each VN in most trials and the provisioning cost per VN takes almost the same value in both proposed and benchmark models. In the case of $d = 5$, which is shown in Fig. 11(b), Greedy-A and Greedy-B algorithms achieve smaller provisioning cost than the benchmark model; the proposed model provisions VNs more accurately than

TABLE IX
STANDARD DEVIATIONS OF COMPUTATION TIME [S] CORRESPONDING TO FIG. 10(A) ($|R| = 1$ AND $d = 1$).

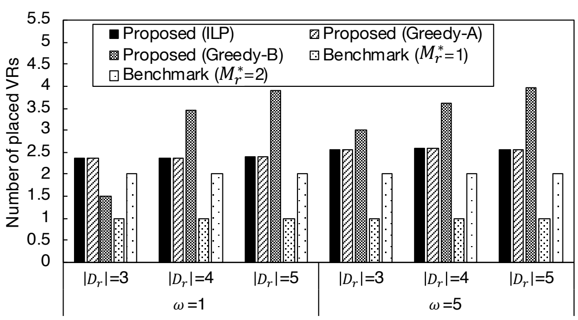
	$\omega = 1$			$\omega = 5$		
	$\lvert D_r \rvert = 3$	$\lvert D_r \rvert = 4$	$\lvert D_r \rvert = 5$	$\lvert D_r \rvert = 3$	$\lvert D_r \rvert = 4$	$\lvert D_r \rvert = 5$
Proposed (ILP)	5.63	852.36	2005.22	216.56	1021.60	3006.92
Proposed (Greedy-A)	20.93	2890.72	9669.61	246.30	2596.33	11998.80
Proposed (Greedy-B)	0.46	1.43	12.19	0.42	6.38	67.68
Benchmark ($M_r^* = 1$)	0.29	0.43	0.91	0.27	0.44	0.95
Benchmark ($M_r^* = 2$)	322.95	1833.84	3445.38	268.92	928.41	1867.37

TABLE X
STANDARD DEVIATIONS OF COMPUTATION TIME [S] CORRESPONDING TO FIG. 10(B) ($|R| = 1$ AND $d = 10$).

	$\omega = 1$			$\omega = 5$		
	$\lvert D_r \rvert = 3$	$\lvert D_r \rvert = 4$	$\lvert D_r \rvert = 5$	$\lvert D_r \rvert = 3$	$\lvert D_r \rvert = 4$	$\lvert D_r \rvert = 5$
Proposed (ILP)	181.77	724.87	2506.52	226.39	1249.37	2702.48
Proposed (Greedy-A)	388.14	1875.91	10497.65	246.71	2602.50	12069.85
Proposed (Greedy-B)	0.55	11.08	94.45	0.87	7.86	79.64
Benchmark ($M_r^* = 1$)	0.24	0.46	0.66	0.31	0.49	0.72
Benchmark ($M_r^* = 2$)	115.50	573.55	2252.01	356.14	762.77	2064.11

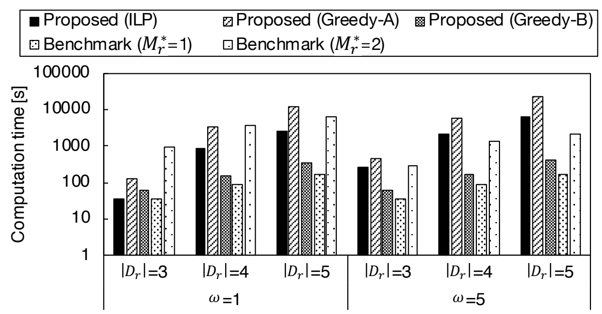


(a) $d = 1$.

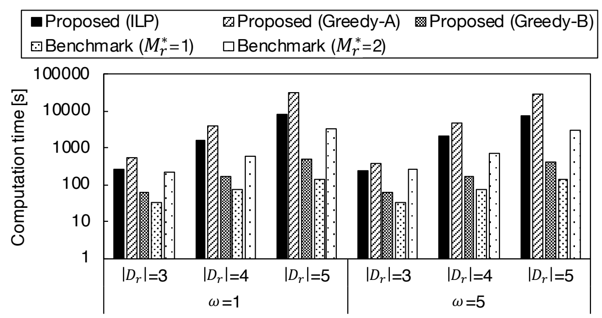


(b) $d = 10$.

Fig. 9. Average number of placed VRs depending on number of DCs ($|R| = 1$).



(a) $d = 1$.



(b) $d = 10$.

Fig. 10. Average computation time depending on number of DCs ($|R| = 1$).

the benchmark model. Compared with the benchmark model, Greedy-A and Greedy-B reduce the provisioning cost by at most 28.8% and 22.6%, respectively. The difference between the provisioning costs of Greedy-A and Greedy-B is at most 8.2% in our examined scenarios. The Greedy-A and Greedy-B can place the different numbers of VRs, as shown in Fig. 12.

Fig. 13 shows the average computation time to obtain VN provisioning when the number of VNs is set to $2 \leq |R| \leq 5$. Tables XI and XII show the standard deviations corresponding to the results in Figs. 13(a) and 13(b), respectively. The computation time increases in proportion to $|R|$. Fig. 13(a) shows that the computation time of the ILP approach exceeds that of Greedy-A when $|D_r| = 5$ and $|R| \geq 3$; Greedy-A can be an attractive option in this range of parameters. Greedy-B runs about 20–64 times faster than Greedy-A in our examined scenarios, with less computation time deviation.

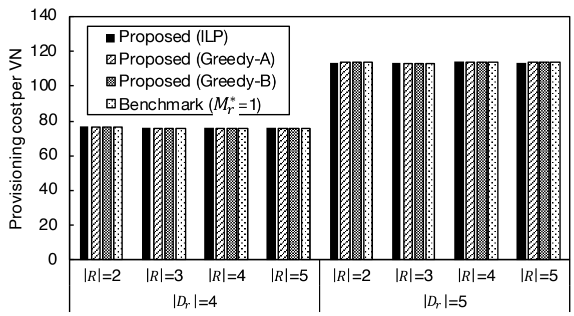
Network operators can choose an adequate approach in the proposed model based on the conditions of VN requests and the limitation of computation time. As discussed in Section IV, the ILP problem solved in Greedy-A has less constraints and decision variables than the original ILP problem; Greedy-A requires less memory for solving the problem. Greedy-A achieves smaller provisioning cost compared to Greedy-B in several settings where the number of VNs is set to $2 \leq |R|$, as shown in Fig. 11(b). In such settings, Greedy-A can be

TABLE XI
STANDARD DEVIATIONS OF COMPUTATION TIME [S] CORRESPONDING TO FIG. 13(A) ($|D_r| = 4, 5$ AND $d = 1$).

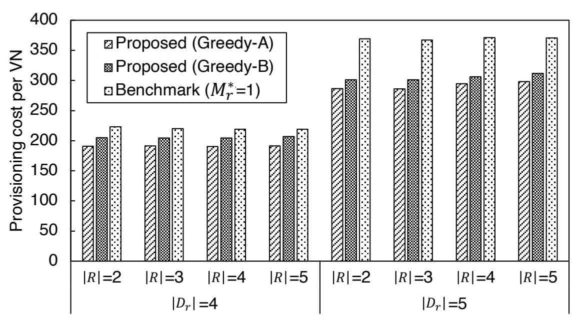
	$\lvert D_r\rvert = 4$				$\lvert D_r\rvert = 5$			
	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$	$\lvert R\rvert = 4$	$\lvert R\rvert = 5$	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$	$\lvert R\rvert = 4$	$\lvert R\rvert = 5$
Proposed (ILP)	1188.46	1756.72	2308.32	2826.72	17962.08	78049.73	92412.24	367355.88
Proposed (Greedy-A)	3487.94	4075.05	4884.19	3596.09	12606.20	15538.23	17554.19	18975.77
Proposed (Greedy-B)	1.73	3.38	3.20	3.45	18.97	22.22	25.51	25.73
Benchmark ($M_r^* = 1$)	3.17	4.76	5.77	8.49	0.93	1.44	1.48	2.69

TABLE XII
STANDARD DEVIATIONS OF COMPUTATION TIME [S] CORRESPONDING TO FIG. 13(B) ($|D_r| = 4, 5$ AND $d = 5$).

	$\lvert D_r\rvert = 4$				$\lvert D_r\rvert = 5$			
	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$	$\lvert R\rvert = 4$	$\lvert R\rvert = 5$	$\lvert R\rvert = 2$	$\lvert R\rvert = 3$	$\lvert R\rvert = 4$	$\lvert R\rvert = 5$
Proposed (Greedy-A)	4323.77	4526.76	5069.38	5821.23	8982.46	21992.53	41297.72	42503.66
Proposed (Greedy-B)	19.59	26.82	33.78	35.57	107.20	166.82	218.81	229.55
Benchmark ($M_r^* = 1$)	3.81	5.23	4.97	9.65	1.02	2.10	1.83	3.37

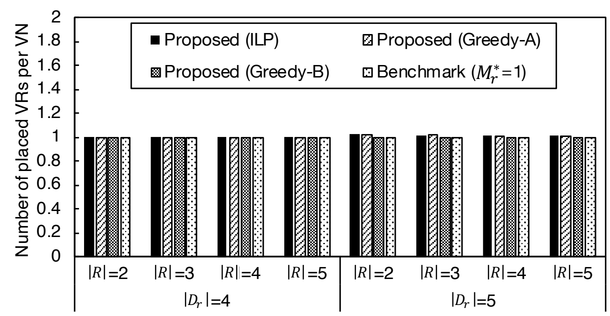


(a) $d = 1$.

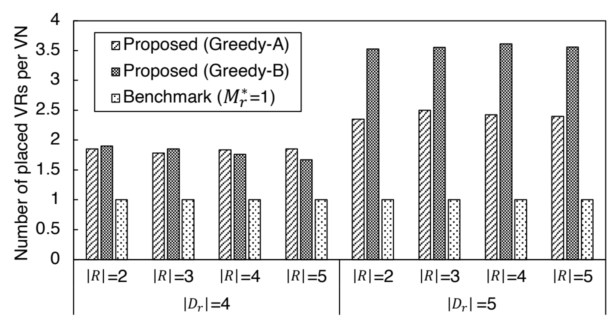


(b) $d = 5$.

Fig. 11. Average provisioning cost per VN for various numbers of VN requests ($|D_r| = 4, 5$).



(a) $d = 1$.



(b) $d = 5$.

Fig. 12. Average number of placed VRs per VN for various numbers of VNs ($|D_r| = 4, 5$).

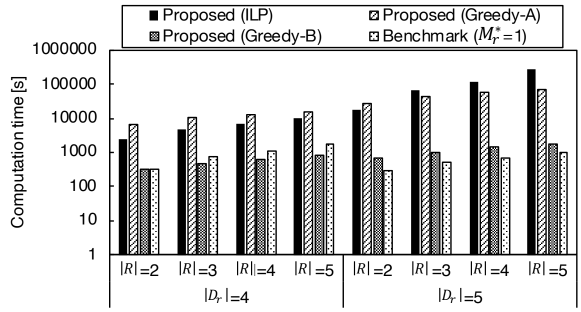
a valuable option for obtaining the VN provisioning. The results in Figs. 10 and 13 indicate that the computation time depends on the decision variables' cardinalities. The network operator can refer to historic runtimes to select an algorithm. A practical scenario to which the VNDE model is adopted is that a network operator periodically updates VNs according to the solution of the VNDE problem. If the expecting runtime of solving the ILP problem is longer than the period of updates, Greedy-B needs to be used.

4) Demonstration with different objective function: In this section, we demonstrate the performance of the proposed model by using the load-balancing function as the objective function. Inspired by one of the objective functions presented in [18], which is named "load balancing plus ε shortest path (LB+εSP)," we set a new objective function for the VNDE model as:

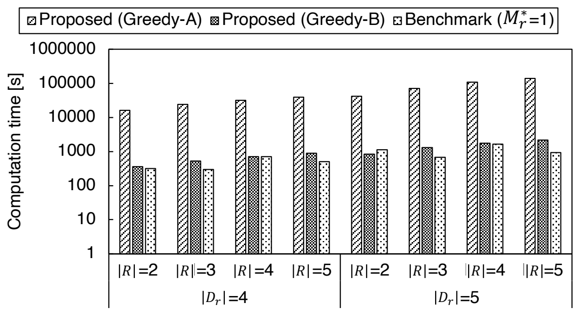
$$\min \quad \alpha^{\mathrm{N}} \mathcal{L}^{\mathrm{N}}_{\max} + \alpha^{\mathrm{L}} \mathcal{L}^{\mathrm{L}}_{\max} + \varepsilon \sum_{(i,j)\in E^{\mathrm{S}}} \mathcal{B}_{ij}, \tag{27a}$$

where

$$\sum_{r\in R} \sum_{(p,q)\in P_r} d_{rpq} \Bigg(\sum_{w\in N^{\mathrm{V}}:(v,w)\in E^{\mathrm{V}}} \xi^{rpq}_{vw} + \sum_{w\in N^{\mathrm{V}}:(w,v)\in E^{\mathrm{V}}} \xi^{rpq}_{wv} + \sum_{(i,j)\in E^{\mathrm{S}}:j=v} \theta^{rpv}_{ij,\mathrm{UP}} + \sum_{(i,j)\in E^{\mathrm{S}}:i=v} \theta^{rvq}_{ij,\mathrm{DW}} \Bigg) \leq c^{\mathrm{N}}_v \mathcal{L}^{\mathrm{N}}_{\max}, \forall v \in N^{\mathrm{V}}, \tag{27b}$$



(a) $d = 1$.



(b) $d = 5$.

Fig. 13. Average computation time for various numbers of VNs ($|D_r| = 4, 5$).

$$\mathcal{B}_{ij} = \sum_{r \in R} \sum_{(p,q) \in P_r} d_{rpq} \left(\sum_{(v,w) \in E^{\mathrm{V}}} \mu^{rpq}_{vwij} + \sum_{v \in N^{\mathrm{V}}} \left(\theta^{rpv}_{ij,\mathrm{UP}} + \theta^{rvq}_{ij,\mathrm{DW}} \right) \right), \forall (i,j) \in E^{\mathrm{S}}, \tag{27c}$$

and

$$\mathcal{B}_{ij} \leq c^{\mathrm{L}}_{ij} \mathcal{L}^{\mathrm{L}}_{\max}, \forall (i,j) \in E^{\mathrm{S}}. \tag{27d}$$

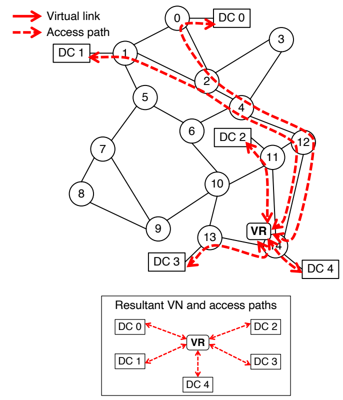
$\mathcal{L}^{\mathrm{N}}_{\max}$ and $\mathcal{L}^{\mathrm{L}}_{\max}$ are positive decision variables that denote the maximum loads of routers and substrate links, respectively. $\mathcal{B}_{ij}$ is a positive decision variable that is equivalent to the left-hand side of (7b); it denotes the amount of all traffic that passes through substrate link $(i,j) \in E^{\mathrm{S}}$. α^{N} and α^{L} are given cost-coefficient parameters that weight the loads of routers and substrate links, respectively. ε is a small constant which is introduced to prioritize the first and second terms over the third term in (27a).

We set $\alpha^{\mathrm{N}} = \alpha^{\mathrm{L}} = 0.5$ and $\varepsilon = 1.0 \times 10^{-8}$ in this evaluation. We set $d = 10$, $\omega = 1$, $|D_r| = 5$, and $|R| = 1$. The maximum computation time for solving an ILP problem is set to 12 hours. Other simulation conditions are the same as those described in Section VI-A1.

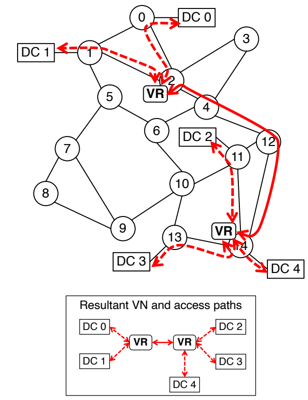
Table XIII shows the average objective value of five trials obtained by the proposed model and the benchmark algorithm with $M_r^* = 1$. It can be observed that the proposed model achieves a smaller objective value than the benchmark algorithm, which indicates that the proposed model achieves load balancing effectively. Fig. 14 shows examples of resultant VNs, which are obtained in one of the trials in this evaluation. Customer's DCs are attached to substrate routers 0, 1, 11, 13, and 14. Fig. 14(a) shows that, when the benchmark algorithm is used, only one VR is placed on the network. All data traffic originating from or destined to DC 0 and DC 1 passes through substrate links (2, 4), (4, 12), and (12, 14); $\mathcal{B}_{ij} = \mathcal{L}^{\mathrm{L}}_{\max} = 80$ Gbps in these links. On the other hand, as shown in Fig. 14(b), the proposed model places two VRs on the network. Data traffic between DC 0 and DC 1 is directly exchanged via the VR deployed in router 2, which reduces the traffic amount on substrate links (2, 4), (4, 12), and (12, 14) to $\mathcal{B}_{ij} = \mathcal{L}^{\mathrm{L}}_{\max} = 60$ Gbps.

TABLE XIII
RESULTS OF VNDE WITH LOAD-BALANCING OBJECTIVE FUNCTION.

	Benchmark ($M_r^* = 1$)	Proposed
Average objective value	0.44	0.39



(a) Benchmark algorithm ($M_r^* = 1$).



(b) Proposed model.

Fig. 14. Examples of resultant VNDE with load-balancing objective function.

B. Large-size networks

1) Simulation environment: We evaluate the performance of the proposed VNDE model by using substrate networks which have larger scale than the Atlanta network. Fig. 15 shows the substrate networks used in this evaluation: (a) US Backbone [56], which has 24 nodes and 86 directed links, (b) Germany50 [54], which has 50 nodes and 176 directed links, and (c) SINET5 [57], which has 50 nodes and 158 directed links[2]. The capacity of each substrate link, c^{L}_{ij}, is set to 100 Gbps. We assume that all substrate routers can be VRs, i.e., $N^{\mathrm{V}} = N^{\mathrm{S}}$. A set of logical links, E^{V}, is selected as follows:

- In (a) US backbone, a logical link is established between every pair of substrate routers.
- In (b) Germany50, a logical link is set between every pair of substrate routers, where the shortest path distance between the routers is not more than 540 km.
- In (c) SINET5, a logical link is set between every pair of substrate routers, where the shortest path distance between the routers is not more than 540 km[3]. A logical

[2]The length of each substrate link is set by assuming that each substrate router is located in a government building of each prefecture in Japan.

[3]This is equal to the shortest path distance between routers 10 and 25 in Fig. 15(c).

link is also established for each of router pairs (0, 3), (41, 47), and (45, 47)[4].

The capacity of each router, c_v^{N}, is set to 1000 Gbps for (a) US backbone and 100 Gbps for the other two networks. We set 14 cost classes for transit VRs, VLs, and access VRs. The utilization costs of a VR and a VL are set as shown in Figs. 16(a) and 16(b), respectively. The maximum allowable number of VRs in each VN, M_r, is set to the number of DCs that belong to VN request $r \in R$. The maximum allowable delay of each source-destination DC pair, κ_{rpq}, is set to 20000 time units for (a) US backbone and 10000 time units for the other two networks. The delay at each VR, β_v, is set to 200 time units. The delay of each substrate link, γ_{ij}, depends on the length of each link; we use 1 time unit/km to set γ_{ij} in this simulation.

We generate 84 VN requests, which consist of 56 VNs requesting 2 DCs, 17 VNs requesting 3 DCs, 7 VNs requesting 4 DCs, and 4 VNs requesting 5 DCs, based on the data of sub-campus numbers accommodated by SINET5 [57]. A substrate router to which each DC is connected is randomly selected from among all substrate routers such that the selected routers do not overlap among the DCs. The traffic amount generated by each DC is randomly selected from the range of $0.01 \leq d_{\text{total}} \leq 10$ Gbps, where a cumulative distribution function is $y = 0.57 \times (100 d_{\text{total}})^{0.081}$, based on the data of measured access link traffic in SINET5 [57]. We assume that a DC sends the same amount of traffic to all other DCs in the same VN. For example, if the traffic amount generated by DC $p \in D_r$ is d_{total} Gbps, DC p sends $d_{rpq} = \frac{d_{\text{total}}}{D_r - 1}$ Gbps to each of the other DCs in VN r. The utilization cost of substrate link per unit transmission capacity, ω_{ij}, depends on the link length; we use 1, 2, 5, and 10 cost units/km/Gbps to set ω_{ij} in this simulation. We obtain the VN provisioning by running Greedy-B.

Simulations in this section use a server with an AMD EPYC 7502P processor with 128 GB memory. ILP problems are solved by CPLEX 20.1.0.0 [55].

2) Results: Tables XIV, XV, and XVI show the performance results of VN provisioning in (a) US Backbone, (b) Germany50, and (c) SINET5, respectively. Fig. 17 shows the distribution of the number of VRs forming each VN in (c) SINET5. Note that we set $\alpha_v^{\mathrm{VR}} = 1$, $\alpha_{vw}^{\mathrm{VL}} = 1$, and $\alpha_{rpv}^{\mathrm{AC}} = 1$ in this evaluation. The total provisioning cost increases as the utilization cost of substrate link, which affects the access path cost, increases. For all networks, the total provisioning cost does not simply increase in proportion to the utilization cost of substrate link. This indicates that the proposed model tends to suppress the provisioning cost by using more VRs as ω_{ij} becomes large, in the same way as shown in Section VI-A3. In the case of (a) US Backbone, which has longer links than the other two networks, more VRs tend to be placed to suppress the link utilization cost; nearly the maximum number of VRs (211 VRs) are placed when the link utilization cost is set to 2 cost units/km/Gbps or more.

[4]These links correspond to long-distance submarine cables connecting islands in the actual SINET5.

3) Dependency on cost coefficients: With the prospect that the costs of using VRs and VLs can differ depending on real network conditions, we evaluate the performance of Greedy-B by using (c) SINET5 with different settings of cost coefficients. Specifically, we evaluate three additional scenarios: 1) $\alpha_v^{\mathrm{VR}} = 2$, $\alpha_{vw}^{\mathrm{VL}} = \alpha_{rpv}^{\mathrm{AC}} = 0.5$, 2) $\alpha_{vw}^{\mathrm{VL}} = 2$, $\alpha_v^{\mathrm{VR}} = \alpha_{rpv}^{\mathrm{AC}} = 0.5$, and 3) $\alpha_{rpv}^{\mathrm{AC}} = 2$, $\alpha_v^{\mathrm{VR}} = \alpha_{vw}^{\mathrm{VL}} = 0.5$. Table XVII shows the results when ω_{ij} is set to 2 cost units/km/Gbps. The difference in the total provisioning costs among the three scenarios is less than 2.4%. The results show that the total number of VRs decreases when α_v^{VR} increases; this indicates that the proposed model is likely to suppress the provisioning cost by using less VRs. The total provisioning cost increases according to $\alpha_{rpv}^{\mathrm{AC}}$ since the total access VR cost does not change depending on the design and embedding.

4) Dependency on number of DCs: We evaluate the performance of Greedy-B by using (c) SINET5 with different numbers of DCs, $|D_r|$. In this evaluation, we set the traffic demand to $d = 1$ Gbps for all 84 VN requests. We set $\alpha_v^{\mathrm{VR}} = 1$, $\alpha_{vw}^{\mathrm{VL}} = 1$, and $\alpha_{rpv}^{\mathrm{AC}} = 1$.

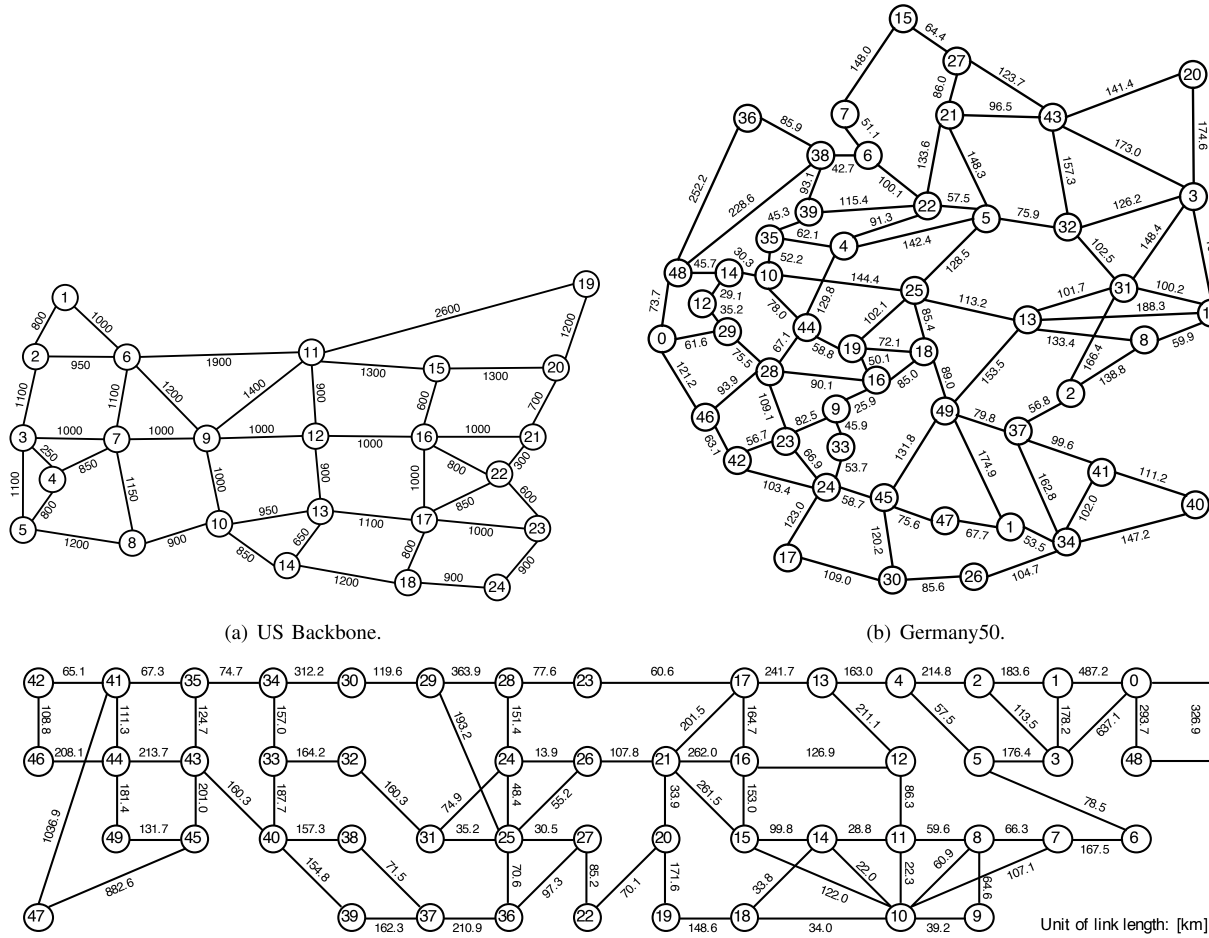
Table XVIII shows the results when $|D_r|$ is varied. Note that we cannot obtain a feasible solution when $|D_r|$ is four or more. The provisioning cost required to provision VNs increases when $|D_r|$ becomes large. The computation time also increases when $|D_r|$ becomes large since more VLs and access paths need to be routed in the network. Fig. 18 shows the distribution of the number of VRs forming each VN. This figure shows that, even if the traffic demand and the number of DCs do not change among VN requests, VNs with different numbers of VRs can be provisioned.

VII. Conclusion

This paper proposed a VNDE model that determines the design and embedding of VNs along with access paths in the single-entity scenario. We formulated the VNDE model as an ILP problem that minimizes the provisioning cost of requested VNs. It sets the utilization costs of transit VR, VL, and access VR as non-decreasing step functions. We developed heuristic algorithms for the case where the ILP problem cannot be solved in practical time. We evaluate the performance of VNDE model by applying it to several networks, including an actual Japanese academic backbone network, SINET5. We demonstrated VN provisioning by using the proposed VNDE model. Numerical results showed that, by using the proposed model, VNs that have an adequate number and locations of VRs, which lead to lowering provisioning cost, can be designed according to the volume of traffic demands and access path cost. The number of VRs placed in the substrate network tends to increase to suppress the provisioning cost when the volume of traffic demands and the utilization cost of substrate links for access paths become large.

References

[1] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.



(a) US Backbone.

(b) Germany50.

(c) SINET5.

Fig. 15. Network models.

TABLE XIV

PERFORMANCE RESULTS OF VN PROVISIONING IN US BACKBONE ($\alpha_v^{\mathrm{VR}} = \alpha_{vw}^{\mathrm{VL}} = \alpha_{rpv}^{\mathrm{AC}} = 1$).

	Utilization cost of substrate link [cost unit/km/Gbps] 1	2	5	10
Total provisioning cost	6.49×10^5	12.31×10^5	29.79×10^5	58.90×10^5
Total number of placed VRs	199	208	208	208
Computation time [s]	11.97×10^3	11.96×10^3	11.99×10^3	12.04×10^3

TABLE XV

PERFORMANCE RESULTS OF VN PROVISIONING IN GERMANY50 ($\alpha_v^{\mathrm{VR}} = \alpha_{vw}^{\mathrm{VL}} = \alpha_{rpv}^{\mathrm{AC}} = 1$).

	Utilization cost of substrate link [cost unit/km/Gbps] 1	2	5	10
Total provisioning cost	1.29×10^5	2.15×10^5	4.52×10^5	8.40×10^5
Total number of placed VRs	134	164	189	197
Computation time [s]	9.84×10^4	9.81×10^4	9.86×10^4	9.91×10^4

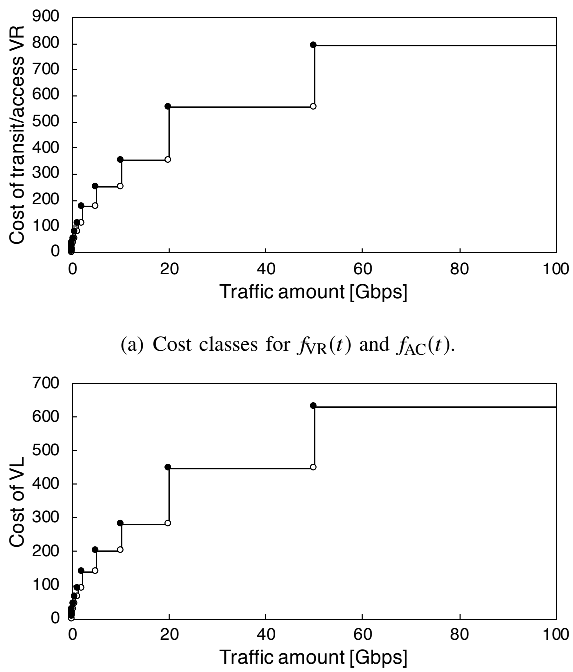
TABLE XVI

PERFORMANCE RESULTS OF VN PROVISIONING IN SINET5 ($\alpha_v^{\mathrm{VR}} = \alpha_{vw}^{\mathrm{VL}} = \alpha_{rpv}^{\mathrm{AC}} = 1$).

	Utilization cost of substrate link [cost unit/km/Gbps] 1	2	5	10
Total provisioning cost	3.04×10^5	5.66×10^5	13.39×10^5	26.22×10^5
Total number of placed VRs	134	158	179	189
Computation time [s]	3.99×10^4	3.99×10^4	3.98×10^4	3.99×10^4

TABLE XVII
PERFORMANCE RESULTS OF VN PROVISIONING IN SINET5 (ω_{ij} IS SET TO 2 COST UNITS/KM/GBPS).

	$\alpha_v^{\mathrm{VR}} = 1$, $\alpha_{vw}^{\mathrm{VL}} = 1$, $\alpha_{rpv}^{\mathrm{AC}} = 1$	$\alpha_v^{\mathrm{VR}} = 2$, $\alpha_{vw}^{\mathrm{VL}} = 0.5$, $\alpha_{rpv}^{\mathrm{AC}} = 0.5$	$\alpha_v^{\mathrm{VR}} = 0.5$, $\alpha_{vw}^{\mathrm{VL}} = 2$, $\alpha_{rpv}^{\mathrm{AC}} = 0.5$	$\alpha_v^{\mathrm{VR}} = 0.5$, $\alpha_{vw}^{\mathrm{VL}} = 0.5$, $\alpha_{rpv}^{\mathrm{AC}} = 2$
Total provisioning cost	5.66×10^5	5.69×10^5	5.57×10^5	5.70×10^5
Total number of placed VRs	158	138	156	177
Computation time [s]	3.99×10^4	3.98×10^4	3.98×10^4	4.00×10^4



(a) Cost classes for $f_{\mathrm{VR}}(t)$ and $f_{\mathrm{AC}}(t)$.

(b) Cost classes for $f_{\mathrm{VL}}(t)$.

Fig. 16. Settings of cost classes in evaluation using large-size networks.

TABLE XVIII
PERFORMANCE RESULTS OF VN PROVISIONING IN SINET5 WITH DIFFERENT DC NUMBERS.

	Number of DCs		
	$\lvert D_r \rvert = 2$	$\lvert D_r \rvert = 3$	$\lvert D_r \rvert \geq 4$
Total provisioning cost	1.51×10^5	4.62×10^5	infeasible
Total number of placed VRs	106	156	
Computation time [s]	1.18×10^4	4.84×10^4	

[2] ETSI, "Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action," Oct. 2012. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[3] M. Veeraraghavan, T. Sato, M. Buchanan, R. Rahimi, S. Okamoto, and N. Yamanaka, "Network function virtualization: A survey," *IEICE Transactions on Communications*, vol. E100-B, no. 11, pp. 1978–1991, Nov. 2017.

[4] IETF, "Automated network management," accessed Mar. 8, 2021. [Online]. Available: https://www.ietf.org/topics/netmgmt/

[5] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Communications Surveys &Tutorials*, vol. 20, no. 4, pp. 3259–3306, Fourth Quarter 2018.

[6] H. Lu, N. Arora, H. Zhang, C. Lumezanu, J. Rhee, and G. Jiang, "HybNET: Network manager for a hybrid network infrastructure," in *Proceedings of the Industrial Track of the 13th ACM/IFIP/USENIX International Middleware Conference*. New York, NY, USA: Association for Computing Machinery, Dec. 2013.

[7] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1114–1128, Fourth Quarter 2012.

[8] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, Fourth Quarter 2013.

[9] 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.

[10] A. Hashmi and C. P. Gupta, "A detailed survey on virtual network embedding," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2019, pp. 730–737.

[11] E. S. Correa, L. A. Fletscher, and J. F. Botero, "Virtual data center embedding: A survey," *IEEE Latin America Transactions*, vol. 13, no. 5, pp. 1661–1670, May 2015.

[12] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, and L. Mathy, "Network virtualization architecture: Proposal and initial prototype," in *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, ser. VISA '09. Association for Computing Machinery, Aug. 2009, pp. 63–72.

[13] R. Bless and C. Werle, "Network virtualization from a signaling perspective," in *2009 IEEE International Conference on Communications Workshops*, Jun. 2009.

[14] National Institute of Informatics, "Science Information NETwork 5," accessed Mar. 8, 2021. [Online]. Available: https://www.sinet.ad.jp/en/top-en

[15] S. Urushidani, S. Abe, K. Yamanaka, K. Aida, S. Yokoyama, H. Yamada, M. Nakamura, K. Fukuda, M. Koibuchi, and S. Yamada, "New directions for a Japanese academic backbone network," *IEICE Transactions on Information and Systems*, vol. E98-D, no. 3, pp. 546–556, Mar. 2015.

[16] T. Sato, T. Kurimoto, S. Urushidani, and E. Oki, "Virtualized network graph design and embedding model," in *IEEE 9th International Conference on Cloud Networking (CloudNet 2020)*, Nov. 2020.

[17] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, Apr. 2006, pp. 1–12.

[18] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, "Optimal virtual network embedding: Node-link formulation," *IEEE Transactions on Network and Service Management*, vol. 10, no. 4, pp. 356–368, Dec. 2013.

[19] R. Mijumbi, J. Serrat, J.-L. Gorricho, and R. Boutaba, "A path generation approach to embedding of virtual networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 334–348, Sep. 2015.

[20] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Apr. 2008. [Online]. Available: https://doi.org/10.1145/1355734.1355737

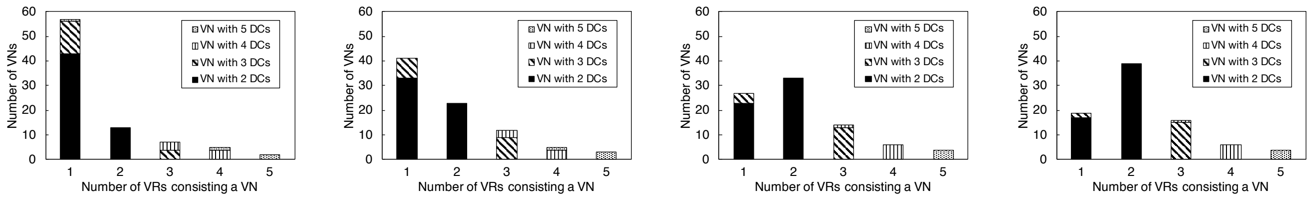
[21] S. Zhang, Z. Qian, J. Wu, S. Lu, and L. Epstein, "Virtual network embedding with opportunistic resource sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 816–827, Mar. 2014.

[22] K. Wen, Z. Qian, and S. Lu, "Breaking the atomicity of virtual network embedding," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 2307–2312.

[23] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on the degree and clustering coefficient information," *IEEE Access*, vol. 4, pp. 8572–8580, Nov. 2016.

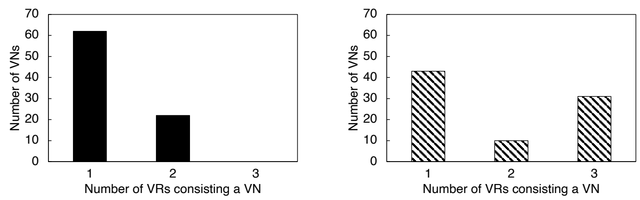
[24] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding LC-VNE algorithms towards integrated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3648–3661, Mar. 2016.

[25] T. He, K.-W. Chin, H. Ren, and X. Liu, "Maximizing virtual network



(a) 1 cost unit/km/Gbps. (b) 2 cost units/km/Gbps. (c) 5 cost units/km/Gbps. (d) 10 cost units/km/Gbps.

Fig. 17. Distribution of number of VRs consisting VN.



(a) $|D_r| = 2$. (b) $|D_r| = 3$.

Fig. 18. Distribution of number of VRs consisting VN with different DC numbers.

embedding requests in RF-charging IoT networks," *IEEE Communications Letters*, vol. 26, no. 4, pp. 863–867, Apr. 2022.

[26] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. de Meer, "Energy efficient virtual network embedding," *IEEE Communications Letters*, vol. 16, no. 5, pp. 756–759, May 2012.

[27] S. Su, Z. Zhang, A. X. Liu, X. Cheng, Y. Wang, and X. Zhao, "Energy-aware virtual network embedding," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1607–1620, Oct. 2014.

[28] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *Journal of Lightwave Technology*, vol. 33, no. 9, pp. 1828–1849, May 2015.

[29] X. Chen, C. Li, and Y. Jiang, "Optimization model and algorithm for energy efficient virtual node embedding," *IEEE Communications Letters*, vol. 19, no. 8, pp. 1327–1330, Aug. 2015.

[30] C. Wang, S. Shanbhag, and T. Wolf, "Virtual network mapping with traffic matrices," in *2012 IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 2717–2722.

[31] R. Lin, S. Luo, J. Zhou, S. Wang, A. Cai, W.-D. Zhong, and M. Zukerman, "Virtual network embedding with adaptive modulation in flexi-grid networks," *Journal of Lightwave Technology*, vol. 36, no. 17, pp. 3551–3563, Sep. 2018.

[32] S. R. Chowdhury, S. Ayoubi, R. Ahmed, N. Shahriar, R. Boutaba, J. Mitra, and L. Liu, "Multi-layer virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1132–1145, Sep. 2018.

[33] K. Ivaturi and T. Wolf, "Mapping of delay-sensitive virtual networks," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, Feb. 2014, pp. 341–347.

[34] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[35] H. Cao, Y. Zhu, G. Zheng, and L. Yang, "A novel optimal mapping algorithm with less computational complexity for virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 356–371, Mar. 2018.

[36] Z. Li, Z. Lu, S. Deng, and X. Gao, "A self-adaptive virtual network embedding algorithm based on software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 362–373, Mar. 2019.

[37] R. Chai, D. Xie, L. Luo, and Q. Chen, "Multi-objective optimization-based virtual network embedding algorithm for software-defined networking," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 532–546, Mar. 2020.

[38] M. Pham, D. B. Hoang, and Z. Chaczko, "Congestion-aware and energy-aware virtual network embedding," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 210–223, Feb. 2020.

[39] H. Lu and F. Zhang, "Resource fragmentation-aware embedding in dynamic network virtualization environments," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 936–948, Jun. 2022.

[40] C. Aguilar-Fuster, M. Zangiabady, J. Zapata-Lara, and J. Rubio-Loyola, "Online virtual network embedding based on virtual links' rate requirements," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1630–1644, Dec. 2018.

[41] C. K. Dehury and P. K. Sahoo, "DYVINE: fitness-based dynamic virtual network embedding in cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1029–1045, May 2019.

[42] H. Cao, S. Wu, G. S. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1406–1416, Feb. 2020.

[43] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNR algorithm: A greedy approach for virtual networks reconfigurations," in *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Dec. 2011, pp. 1–6.

[44] J. Fan and M. H. Ammar, "Dynamic topology configuration in service overlay networks: A study of reconfiguration policies," in *25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, Apr. 2006, pp. 1–12.

[45] H. Yao, S. Ma, J. Wang, P. Zhang, C. Jiang, and S. Guo, "A continuous-decision virtual network embedding scheme relying on reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 864–875, Jun. 2020.

[46] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, Jun. 2020.

[47] D. Andreoletti, T. Velichkova, G. Verticale, M. Tornatore, and S. Giordano, "A privacy-preserving reinforcement learning algorithm for multi-domain virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2291–2304, Dec. 2020.

[48] H. K. Thakkar, C. K. Dehury, and P. K. Sahoo, "MUVINE: Multi-stage virtual network embedding in cloud data centers using reinforcement learning-based predictions," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1058–1074, Jun. 2020.

[49] J. Cheng, Y. Wu, Y. Lin, Y. E, F. Tang, and J. Ge, "VNE-HRL: A proactive virtual network embedding algorithm based on hierarchical reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4075–4087, Dec. 2021.

[50] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: past and future," *IEEE Network*, vol. 16, no. 6, pp. 46–56, Nov. 2002.

[51] S. Huang and R. Dutta, "Dynamic traffic grooming: the changing role of traffic grooming," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 1, pp. 32–50, First Quarter 2007.

[52] B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Communications Surveys& Tutorials*, vol. 17, no. 3, pp. 1776–1800, Third Quarter 2015.

[53] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[54] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0–survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, May 2010.

[55] IBM, "CPLEX optimizer," accessed Mar. 8, 2021. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer

[56] X. Chen, F. Ji, and Z. Zhu, "Service availability oriented p-cycle protection design in elastic optical networks," *Journal of Optical Communications and Networking*, vol. 6, no. 10, pp. 901–910, Oct. 2014.

[57] T. Kurimoto, S. Urushidani, and E. Oki, "Optimization model for designing multiple virtualized campus area networks coordinating with wide area networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1349–1362, Dec. 2018.

Takehiro Sato (S'12–M'16) received the B.E., M.E. and Ph.D. degrees in engineering from Keio University, Japan, in 2010, 2011 and 2016, respectively. He is currently an assistant professor in Graduate School of Informatics, Kyoto University, Japan. From 2011 to 2012, he was a research assistant in the Keio University Global COE Program by Ministry of Education, Culture, Sports, Science and Technology, Japan. From 2012 to 2015, he was a research fellow of Japan Society for the Promotion of Science. From 2016 to 2017, he was a research associate in Graduate School of Science and Technology, Keio University, Japan.

Takashi Kurimoto (M'15) is an Associate Professor with the National Institute of Informatics (NII), Japan. He worked at NTT from 1996 to 2014, where he was engaged in both network planning and research and development of next generation network. In 2015, he joined NII and became involved in the design and implementation of national research and educational networks. His research interests are the switching technology for high-speed computer networks and the next generation network.

Shigeo Urushidani (M'90) is a Deputy Director General and a Professor of the National Institute of Informatics (NII), Japan. He received B.E. and M.E. degrees from Kobe University in 1983 and 1985, respectively, and received a Ph.D. from the University of Tokyo in 2002. He worked for NTT from 1985 to 2006, where he was engaged in the research and development of ATM, AIN, IP/MPLS, and optical switching systems. He moved to NII in 2006 and is currently involved in the design and implementation of the Science Information Network (SINET), as well as in the research on and development of future network architectures.

Eiji Oki (M'95–SM'05–F'13) received the B.E. and M.E. degrees in instrumentation engineering and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1991, 1993, and 1999, respectively. He was with Nippon Telegraph and Telephone Corporation (NTT) Laboratories, Tokyo, from 1993 to 2008, and with the University of Electro-Communications, Tokyo, from 2008 to 2017. From 2000 to 2001, he was a Visiting Scholar at Polytechnic University, Brooklyn, NY. In 2017, he joined Kyoto University, Japan, where he is currently a Professor. His research interests include routing, switching, protocols, optimization, and traffic engineering in communication and information networks.