Achieving a Fully-Flexible Virtual Network Embedding in Elastic Optical Networks

Nashid Shahriar*, Sepehr Taeb*, Shihabur Rahman Chowdhury*, Massimo Tornatore†, Raouf Boutaba*,
Jeebak Mitra‡, and Mahdi Hemmati‡
*David R. Cheriton School of Computer Science, University of Waterloo,
{nshahria | staeb | sr2chowdhury | rboutaba}@uwaterloo.ca
†Politecnico di Milano, massimo.tornatore@polimi.it
‡Huawei Technologies Canada Research Center, {jeebak.mitra | mahdi.hemmati}@huawei.com

***Abstract*—Network operators must continuously scale the capacity of their optical backbone networks to keep apace with the proliferation of bandwidth-intensive applications. Today's optical networks are designed to carry large traffic aggregates with coarse-grained resource allocation, and are not adequate for maximizing utilization of the expensive optical substrate. *Elastic Optical Network (EON)* is an emerging technology that facilitates flexible allocation of fiber spectrum by leveraging finer-grained channel spacing, tunable modulation formats and Forward Error Correction (FEC) overheads, and baud-rate assignment, to right size spectrum allocation to customer needs. Virtual Network Embedding (VNE) over EON has been a recent topic of interest due to its importance for 5G network slicing. However, the problem has not yet been addressed while simultaneously considering the full flexibility offered by an EON. In this paper, we present an optimization model that solves the VNE problem over EON when lightpath configurations can be chosen among a large (and practical) set of combinations of paths, modulation formats, FEC overheads and baud rates. The VNE over EON problem is solved in its splittable version, which significantly increases problem complexity, but is much more likely to return a feasible solution. Given the intractability of the optimal solution, we propose a heuristic to solve larger problem instances. Key results from extensive simulations are: (i) a fully-flexible VNE can save up to 60% spectrum resources compared to that where no flexibility is exploited, and (ii) solutions of our heuristic fall in more than 90% of the cases, within 5% of the optimal solution, while executing several orders of magnitude faster.**

I. Introduction

The proliferation of bandwidth intensive applications such as video on-demand, online social media, *etc.*, is resulting in an unprecedented growth of the global Internet traffic. For instance, AT&T experienced 100,000% increase in traffic between 2008 and 2016 [1], and Cisco forecasts the global Internet traffic to triple by 2021 [2]. As a consequence, network operators are constantly faced with the challenge to scale capacity of their optical backbone networks. Fixed-grid Dense Wavelength Division Multiplexing (DWDM) is the *de facto* technology for deploying optical networks. It multiplexes multiple client signals on the same fiber by dividing the frequency spectrum into a large fixed-sized grid (*e.g.*, typically 50GHz wide slices, each supporting 100Gbps data-rate with standard modulation format [3]) and mapping a client signal to a single slice in the grid. Although DWDM increases fiber utilization, it is falling short of meeting today's capacity requirements for multiple reasons. First, coarse-grained spectrum allocation in DWDM leads to wasted capacity, *e.g.*, fitting a 40Gbps Ethernet signal wastes 60% of slice capacity. Second, fixed-grid DWDM is not flexible enough to meet the growing need for supporting data-rates beyond 100Gbps [3].

The shortcomings of fixed-grid DWDM have triggered the evolution towards flex-grid optical networks. The more recent flex-grid approach divides the frequency spectrum into smaller (*e.g.*, 12.5Ghz) slices and allows allocation of an arbitrary number of slices to right size spectrum allocation to customer needs. Flex-grid, combined with the recent advances in coherent-transmission technology, can bring unprecedented elasticity in tuning transmission parameters such as baud rate, modulation format and forward error correction (FEC) overhead. In addition, emerging transponder technologies such as bandwidth variable transponders and multi-flow transponders offer unprecedented capability of transmitting and receiving a traffic demand into multiple elastic optical flows, operating at independent bit rates, that can be routed independently [4]. This new generation of optical networks is usually referred to as Elastic Optical Networks (EONs) [3], [5].

EON virtualization is currently gaining interest, especially because of its importance in future 5G network slicing [5]. A core problem in EON virtualization is to efficiently map a virtual network on the substrate optical network, also known as the Virtual Network Embedding (VNE) problem [6]. VNE has been extensively studied for layer-2/3 electrical and fixed-grid optical networks [6]. However, VNE over EON introduces unique challenges and complexities due to the large number of flexible transmission parameters made available by EONs: (i) fine-grained spectrum allocation, (ii) baud-rate selection (*e.g.*, 28, 47.5, 66 Gbaud/s *etc.*), (iii) adaptive modulation formats (*e.g.*, ranging from BPSK to 128QAM [7]), and (iv) adaptive FEC overhead (*e.g.*, 7%, 20%, or even more as in [8]). Hence, unlike layer-2/3 or fixed-grid networks, virtual link mapping in EON is not merely allocating bandwidth or selecting a continuous wavelength on a path, respectively, which are known to be NP-hard [9], [10]. For mapping even a single virtual link, EON mandates selecting transmission parameters from a combinatorial set of options, which in-turn determines the usability of an exponential number of paths for mapping.

The state-of-the-art in VNE over EON does not jointly consider all the aforementioned flexible transmission parameters, rather considers only a subset of them [11], [12]. Another aspect of the problem is cost-benefit analysis, *i.e.*, what is the cost of introducing flexibility in a specific transmission parameter (*e.g.*, upgrading transceivers to support adaptive modulation formats and FECs) and how much are the resource allocation benefits for VNE stemming from this flexibility. In this paper, we attempt to fill the gap by answering the following questions: *(i) how does jointly considering all the flexible transmission parameters impact resource allocation for VNE? (ii) what is the gain in resource allocation as a result of incrementally introducing flexibility for VNE over EON? (e.g., upgrading only the transceivers to enable adaptive modulation format vs. upgrading the transceivers and also the optical cross-connects to enable flexible spectrum allocation)?*

To answer these questions, we develop a mathematical model to optimally solve VNE on EON by jointly considering all the flexible transmission parameters, while minimizing the total spectrum usage (Section III and Section IV). Note that we do not consider the virtual nodes to be already placed, and allow to split virtual link demands on multiple substrate paths similar to [4]. This is particularly useful when no substrate path can yield a data-rate large enough to accommodate a virtual link's demand. Given the NP-hardness of the optimal solution, we propose a heuristic to solve larger instances of the problem (Section V). To the best of our knowledge, this is the first optimization model capturing flexibility in all the transmission parameters of an EON, while considering virtual link splitting. The optimization model allows us to consider the impact of flexibility in one specific transmission parameter (*e.g.*, modulation format) while fixing the others (*e.g.*, fixed FEC overhead and no flexible spectrum allocation) to quantify its impact on resource allocation for VNE. Extensive simulations using realistic network topologies from [13] demonstrate the spectrum savings achieved by a fully-flexible VNE and the efficiency of our heuristic solution (Section VI).

II. Related Works

Several variants of the VNE problem have been studied over the last decade for layer 2/3 network virtualization [6]. The problem has been investigated in the context of traditional optical networks [7] and, more recently, in the context of EONs [11], [12], [14]. The authors in [11] and [12] investigated two variants of the VNE problem over EONs, namely transparent and opaque VNE. In the transparent case all the virtual links of a VN must use the same spectrum for their substrate links, while in the opaque VNE case only the substrate links in the embedding of a virtual link must use the same spectrum. While [11] and [12] proposed a formulation based on multi-commodity flow, [14] uses pre-computed path based formulations for the VNE over EON problem.

However, none of these works leverage the full flexibility of an EON, namely combining different modulation formats, FECs, and baud rates to allow multiple transmission reaches. Several works have considered the impact of these transmission parameters in isolation, and not in the context of the VNE problem (see e.g., [15] for distance-adaptive modulation formats, [16] for variable baud rates, and [8], [17] for variable FEC redundancy). The joint effect of modulation formats and FEC redundancy was only considered in [18], where authors show that, by adapting both modulation format and FEC, one can significantly decrease resource occupation compared to the cases when modulation-format and FEC adaptation are considered in isolation. But [18] only provides a simulation study in a dynamic environment, and does not formulate and solve the VNE problem over EON with full flexibility in a static scenario as we do in this study.

None of the above approaches considers the possibility of splitting a virtual link demand on multiple substrate paths. This poses significant limitations as transmission reach/spectrum contiguity/capacity constraints may inhibit all the available paths in the EON to serve high bandwidth demands, thus blocking the VN request. The idea of splitting requests has been investigated in EONs as split-spectrum-enabled EONs [4], but not in the context of VNE, and not considering full flexibility. Our study provides novel contributions to the VNE over EON problem by: i) solving the problem under assumption of full flexibility in terms of transmission parameters (modulation formats, FEC, baud rates) and ii) allowing the formulation to split requests with high data-rates.

III. Problem Statement and Notation

A. Substrate EON

The substrate EON (SN) is represented as an undirected graph $G = (V, E)$, where V and E are the set of substrate optical nodes (SNodes) and substrate optical links (SLinks), respectively. Without loss of generality, we assume the optical nodes to be colorless, directionless, and contentionless [19]. We also assume the existence of two optical fibers in each directions between adjacent optical nodes, therefore, consider the SLinks to be bi-directional. The optical frequency spectrum on each SLink $e = (u, v) \in E$ is divided into equal width frequency slices (we use the term frequency/spectrum slot and slice interchangeably) represented by the set S and enumerated as $1, 2 \ldots |S|$. $\mathcal{P}$ and $\mathcal{P}_{uv}^{k} \subset \mathcal{P}$ represents the set of all paths in G and the set of k-shortest paths between nodes $u, v \in V$, respectively. The endpoints of a path $p \in \mathcal{P}$ are denoted by $src(p)$ and $dst(p)$. We denote the association between a path $p \in \mathcal{P}$ and any link $e \in E$ by the binary variable δ_{pe}, which is set to 1 when e is on p, 0 otherwise.

A physical path p with length l_p can be configured with the following parameters to enable data transmission with different data-rates $d \in \mathcal{D}$: *baud-rate*, b, *modulation format*, m, and *FEC*, f, selected from the set of possible values $\mathcal{B}$, $\mathcal{M}$, and $\mathcal{F}$, respectively. We use a tuple $t = (d, b, m, f) \in \mathcal{T} = (\mathcal{D} \times \mathcal{B} \times \mathcal{M} \times \mathcal{F})$ to represent a transmission configuration that dictates the combination of $b \in \mathcal{B}$, $m \in \mathcal{M}$, and $f \in \mathcal{F}$ that can be used to yield a data-rate $d \in \mathcal{D}$. For the sake of representation, we use $t^{(d)}, t^{(b)}, t^{(m)}$, and $t^{(f)}$ to denote the data-rate, baud-rate, modulation format, and FEC of a configuration $t \in \mathcal{T}$. A *reach table* $\mathcal{R}$, computed based on physical layer characteristics, specifies the maximum length of a path (*i.e.*, the *reach* r_t) capable of retaining a satisfactory signal quality when configured according to a transmission configuration $t \in \mathcal{T}$. Finally, n_t denotes the number of slices required to accommodate a transmission configuration $t \in \mathcal{T}$, which is dependent on the parameters of t.

B. Virtual Network Request

The virtual network request is represented by an undirected graph $\bar{G} = (\bar{V}, \bar{E})$, where $\bar{V}$ and $\bar{E}$ are the set of virtual nodes (VNodes) and virtual links (VLinks), respectively. Each VNode $\bar{u} \in \bar{V}$ is associated with a location constraint set $\mathcal{L}(\bar{u}) \subset V$ indicating the set of SNodes where $\bar{v}$ can be embedded. We use the following binary variable to represent location constraints: $\ell_{\bar{u}u} = 1$ if $u \in V$ is in the location constraint set $\mathcal{L}(\bar{u})$, 0 otherwise. We represent the endpoints of a VLink $\bar{e} = (\bar{u}, \bar{v}) \in \bar{E}$ by $src(\bar{e})$ and $dst(\bar{e})$. Each VLink $\bar{e} \in \bar{E}$ has a bandwidth requirement $\bar{\beta}_{\bar{e}}$. We assume the VLinks can be mapped to one or more substrate paths (SPaths), each with a equal or lower data-rate than $\bar{\beta}_{\bar{e}}$. This allows to embed VLinks with higher data-rates (*e.g.*, more than 400Gbps) for which the reach becomes smaller limiting the number of usable paths. However, we limit the number of VLink splits to maximum q (≥ 1) SPaths.

C. Pre-computations

For each VLink $\bar{e} \in \bar{E}$, we pre-compute $\mathcal{P}_{\bar{e}}^{k}$, a set of k shortest paths between each pair of SNodes in the VLink's endpoints' location constraint sets. For each SPath $p \in \mathcal{P}_{\bar{e}}^{k}$, we pre-compute the set of admissible transmission configurations, $\mathcal{T}_{\bar{e}p} \subset \mathcal{T}$, such that each configuration $t \in \mathcal{T}_{\bar{e}p}$ results in a reach $r_t \geq l_p$ and has a data-rate $t^{(d)}$. $\mathcal{T}_{\bar{e}}$ contains all the distinct tuples suitable for $\bar{e}$ and is defined as $\bigcup_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \mathcal{T}_{\bar{e}p}$.

D. Problem Statement

Given an SN G, a reach table $\mathcal{R}$, and a VN request $\bar{G}$ with location constraint set $\mathcal{L}$:

- Compute the node embedding function $\tau : \bar{V} \to V$, *i.e.*, embed each VNode $\bar{u} \in \bar{V}$ to an SNode $u \in \mathcal{L}(\bar{u})$.
- Compute the link embedding function $\gamma : \bar{E} \to \chi : \chi \subset \mathcal{P} \times \mathcal{T} \times S^2$ and $1 \leq |\chi| \leq q$, *i.e.*, compute up to a maximum of q splits for the bandwidth demand $\bar{\beta}_{\bar{e}}$ of each VLink $\bar{e} \in \bar{E}$, select an SPath and an appropriate transmission configuration $t \in \mathcal{T}$ from the reach table $\mathcal{R}$ for each split, and allocate a contiguous segment of slices represented by the starting and ending slice index on each SLink along the SPath. $\chi_{\bar{e}i} = (p, t, s_b, s_t) | 1 \leq i \leq q$ represents the i-th split, where $\chi_{\bar{e}i}^{(p)}$ and $\chi_{\bar{e}i}^{(t)}$ denote the selected SPath and transmission configuration for the i-th split, respectively. In addition, allocation of spectrum slices for the i-th split begins at index $\chi_{\bar{e}i}^{(s_b)}$ and ends at index $\chi_{\bar{e}i}^{(s_t)}$ along each SLink in the SPath $\chi_{\bar{e}i}^{(p)}$.
- The total number of slices required to provision the VN is minimum according to the following cost function:
$$\sum_{\bar{e} \in \bar{E}} \sum_{i=1}^{q} (\chi_{\bar{e}i}^{(s_t)} - \chi_{\bar{e}i}^{(s_b)} + 1) \times |\chi_{\bar{e}i}^{(p)}| \tag{1}$$
where $|\chi_{\bar{e}i}^{(p)}|$ denotes the number of SLinks in $\chi_{\bar{e}i}^{(p)}$.

The above are subject to spectrum resource constraints (*i.e.*, cannot allocate more than the available number of slices on a link and a slice on a link is assigned to at most one split) and spectral contiguity (*i.e.*, consecutive slices of a split are adjacent to each other) and continuity (*i.e.*, for a split, the same set of slices are allocated on each link of a path) constraints.

IV. Problem Formulation

We present a path-based formulation for the VNE over EON problem, *i.e.*, we provide the set of possible paths that can be used to embed VLinks as in input to the formulation.

A. Decision Variables

A VNode is mapped to exactly one SNode from its location constraint set, denoted by the following decision variable:

$$x_{\bar{u}u} = \begin{cases} 1 & \text{if } \bar{u} \in \bar{V} \text{ is mapped to } u \in V \\ 0 & \text{otherwise} \end{cases}$$

We allow VLink splitting for embedding, *i.e.*, satisfy the demand of a VLink by provisioning slices over multiple SPaths up to a maximum of q SPaths as q is the maximum number of allowable splits. Slice provisioning over multiple SPaths can be done using a combination of the following two cases: (i) allocate contiguous sets of slices on distinct SPaths; (ii) allocate different contiguous sets of slices on the same SPath (with identical or different transmission configuration). The latter represents a scenario when there are sufficient slices available over an SPath, however, the SPath length does not allow provisioning the total required data-rate using a single contiguous set of slices. To represent the second case, we assume each transmission configuration for an SPath can be instantiated multiple times (up to a maximum of q times as total number of splits cannot exceed q regardless of the case). The following variable relates a VLink to an SPath and an instance of a transmission configuration on that SPath:

$$w_{\bar{e}pti} = \begin{cases} 1 & \text{if } \bar{e} \in \bar{E} \text{ uses } i\text{-th instance of } t \in \mathcal{T}_{\bar{e}p} \\ & \quad \text{on path } p \in \mathcal{P}_{\bar{e}}^{k} \\ 0 & \text{otherwise} \end{cases}$$

Finally, the following decision variable creates the relationship between a mapped SPath and the slices in its links:

$$y_{\bar{e}ptis} = \begin{cases} 1 & \text{if } \bar{e} \in \bar{E} \text{ uses slice } s \in S \text{ on path } p \in \mathcal{P}_{\bar{e}}^{k} \\ & \text{with the } i\text{-th instance of } t \in \mathcal{T}_{\bar{e}p} \\ 0 & \text{otherwise} \end{cases}$$

B. Constraints

a) VNode Mapping Constraints: We ensure by (2) and (3) that each VNode is mapped to an SNode satisfying the location constraints. Then, (4) constraints an SNode to host at most one VNode from the same VN. Note that VNode mapping follows from VLink mapping, since there is no cost associated with the VNode mapping.

$$\forall \bar{u} \in \bar{V}, \forall u \in V : x_{\bar{u}u} \leq \ell_{\bar{u}u} \tag{2}$$

$$\forall \bar{u} \in \bar{V} : \sum_{u \in V} x_{\bar{u}u} = 1 \tag{3}$$

$$\forall u \in V : \sum_{\bar{u} \in \bar{V}} x_{\bar{u}u} \leq 1 \tag{4}$$

b) VLink demand constraints: We provision a VLink by splitting it across multiple (up to q) SPaths. Constraint (5) ensures that for each VLink $\bar{e} \in \bar{E}$, the sum of data-rates resulting from applying the selected transmission configuration on the selected paths is equal to the VLink's demand. Then, (6) enforces an upper limit on the number of splits.

$$\forall \bar{e} \in \bar{E} : \sum_{p \in \mathcal{P}^k_{\bar{e}}} \sum_{t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} (w_{\bar{e}pti} \times t^{(d)}) = \bar{\beta}_{\bar{e}} \tag{5}$$

$$\forall \bar{e} \in \bar{E} : \sum_{p \in \mathcal{P}^k_{\bar{e}}} \sum_{t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} w_{\bar{e}pti} \leq q \tag{6}$$

c) Slice assignment and Spectral Contiguity constraints: We ensure by (7) that if a path p is selected with a specific transmission configuration t, then the required number of slices n_t to support the data-rate $t^{(d)}$ is allocated on the path. (8) ensures that each slice on an SLink is allocated to at most one path. Finally, we ensure the slices allocated on each link of a path are contiguous, *i.e.*, consecutive slices are adjacent to each other in the frequency spectrum by (9).

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}^k_{\bar{e}}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q :$$
$$\sum_{s \in S} y_{\bar{e}ptis} = n_t \times w_{\bar{e}pti} \tag{7}$$

$$\forall e \in E, \forall s \in S :$$
$$\sum_{\bar{e} \in \bar{E}} \sum_{p \in \mathcal{P}^k_{\bar{e}}} \sum_{t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} w_{\bar{e}pti} \times y_{\bar{e}ptis} \times \delta_{pe} \leq 1 \tag{8}$$

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}^k_{\bar{e}}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q, 1 \leq s \leq |S| - 1 :$$
$$\sum_{s'=s+2}^{|S|} y_{\bar{e}ptis'} \leq |S| \times (1 - y_{\bar{e}ptis} + y_{\bar{e}pti(s+1)}) \tag{9}$$

d) Coordination between node and link mapping: (10) ensures that two endpoints of a VLink $\bar{e} \in \bar{E}$ are mapped to the source and destination substrate nodes of an SPath p only when p is used for embedding $\bar{e}$.

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}^k_{\bar{e}}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q :$$
$$x_{src(\bar{e})src(p)} \times x_{dst(\bar{e})dst(p)} \geq w_{\bar{e}pti} \tag{10}$$

(10) is a non-linear constraint. It can be linearized using a technique similar to [12]. We omit the details here for brevity.

1) Objective Function: Our cost function minimizes the total number of spectrum slices required to embed all the VLinks as shown in the first part of (11). However, to break ties among multiple solutions with the same total number of slices, we use the second term in (11) that prefers the solution with the minimum number of splits.

$$\text{minimize}(\sum_{\forall \bar{e} \in \bar{E}} \sum_{p \in \mathcal{P}^k_{\bar{e}}} \sum_{t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} \sum_{s \in S} y_{\bar{e}ptis} +$$
$$\epsilon \times \sum_{\bar{e} \in \bar{E}} \sum_{p \in \mathcal{P}^k_{\bar{e}}} \sum_{t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} w_{\bar{e}pti}) \tag{11}$$

ϵ is chosen in a way that the second term is effective only when multiple solutions have same value for the first part.

V. Heuristic Algorithm

Given the NP-hardness of the ILP formulation, we propose a heuristic to solve larger instances of the VNE over EON problem. In the following, we first discuss how we address the embedding of a single VLink whose endpoints are already mapped (Section V-A). Then we expand our discussion to cover the embedding of a full-fledged VN (Section V-B).

A. Embedding of a single VLink with Mapped Endpoints

Given a VLink with already mapped endpoints, embedding it in an EON involves the following three sub-steps:

a) Path Selection: Recall from Section III-B that a VLink's demand can be satisfied by dividing it into maximum q splits and provisioning each split on an SPath in the EON. Note that different contiguous segments of spectrum slices on the same SPath can be allocated to provision multiple splits on the same SPath. Therefore, the choice of up to q paths from the k shortest paths between a VLink's mapped endpoints for embedding a VLink $\bar{e}$ can be represented using a multi-set $\mathbb{P}^k_{\bar{e}} = (\mathcal{P}^k_{\bar{e}}, m_1)$. $m_1 : \mathcal{P}^k_{\bar{e}} \rightarrow N_q$ (N_q is the set of all integers between 0 and q inclusive) defines the number of times a path $p \in \mathcal{P}^k_{\bar{e}}$ appears in the multi-set $\mathbb{P}^k_{\bar{e}}$. $m_1(p) = 0$ indicates exclusion of a path $p \in \mathcal{P}^k_{\bar{e}}$ from $\mathbb{P}^k_{\bar{e}}$. Size of $\mathbb{P}^k_{\bar{e}}$ is then computed as, $|\mathbb{P}^k_{\bar{e}}| = \sum_{p \in \mathcal{P}^k_{\bar{e}}} m_1(p)$, where $1 \leq |\mathbb{P}^k_{\bar{e}}| \leq q$. As order does not matter in discriminating multisets, $\mathrm{P}^k_{\bar{e}}$ denotes a permutation of the SPaths present in $\mathbb{P}^k_{\bar{e}}$.

b) Transmission Configuration Selection: Each SPath $p \in \mathbb{P}^k_{\bar{e}}$ with $m_1(p) \geq 1$ can be configured with different transmission configurations to provision different data-rates in such a way that altogether they achieve the VLink's demand $\beta_{\bar{e}}$. Recall from Section IV-A that the same transmission configuration can be applied to the same path or different paths multiple times to allocate different contiguous segments of spectrum slices. Therefore, selection of transmission configurations for the paths in $\mathbb{P}^k_{\bar{e}}$ results in a multi-set, $\mathbb{T}_{\mathbb{P}^k_{\bar{e}}} = (\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}, m_2)$, where $\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}$ is the set of feasible transmission configurations for the paths in $\mathbb{P}^k_{\bar{e}}$ and m_2 defines the number of times a transmission configuration appears in $\mathbb{T}_{\mathbb{P}^k_{\bar{e}}}$. There is an one-to-one correspondence between a path in $\mathbb{P}^k_{\bar{e}}$ and a selected transmission configuration in $\mathbb{T}^k_{\mathbb{P}_{\bar{e}}}$, hence, $|\mathbb{T}_{\mathbb{P}^k_{\bar{e}}}| = |\mathbb{P}^k_{\bar{e}}|$.

c) Spectrum Slice Allocation: The final step in VLink embedding is to allocate spectrum slices along each SPath $p \in \mathbb{P}^k_{\bar{e}}$ satisfying constraints (8) and (9). The number of slices to allocate is determined by the transmission configuration $t \in \mathbb{T}_{\mathbb{P}^k_{\bar{e}}}$ selected for p. Slice allocation for a pair $< p, t >$ is represented by the starting slice index $s_b \in S$. The ending slice index for $< p, t >$ can be trivially derived by $s_t = s_b + n_t - 1$, where n_t is the number slices required for t. Since more than one distinct SPaths in $\mathbb{P}^k_{\bar{e}}$ can have the same s_b, the assignment of slices to the SPaths in $\mathbb{P}^k_{\bar{e}}$ is represented as a multi-set of slice indices, $\mathbb{S}_{\mathbb{P}^k_{\bar{e}}} = (S, m_3)$ such that $|\mathbb{S}_{\mathbb{P}^k_{\bar{e}}}| = |\mathbb{P}^k_{\bar{e}}|$, where $m_3 : S_{\mathbb{P}^k_{\bar{e}}} \rightarrow N_q$ defines the number of times a slice index from S appears in $\mathbb{S}_{\mathbb{P}^k_{\bar{e}}}$. Again, there is an one-to-one correspondence between an SPath in $\mathbb{P}^k_{\bar{e}}$ and a slice assignment in $\mathbb{S}_{\mathbb{P}^k_{\bar{e}}}$.

In summary, embedding solution of a VLink $\bar{e}$, $\mathcal{I}_{\bar{e}}$, can be represented as a tuple of three multi-sets, $\mathcal{I}_{\bar{e}} = (\mathbb{P}^k_{\bar{e}}, \mathbb{T}_{\mathbb{P}^k_{\bar{e}}}, \mathbb{S}_{\mathbb{P}^k_{\bar{e}}})$. The optimal embedding for $\bar{e}$, $\mathcal{A}_{\bar{e}}$, is the one requiring minimum number of spectrum slices. If $\mathcal{M}(\mathcal{P}^k_{\bar{e}})$, $\mathcal{M}(\mathcal{T}_{\mathbb{P}^k_{\bar{e}}})$, and $\mathcal{M}(\mathcal{S}_{\mathbb{P}^k_{\bar{e}}})$ are the set of all possible multi-sets of $\mathcal{P}^k_{\bar{e}}$, $\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}$, and

$\mathcal{S}_{\mathbb{P}^k_{\bar{e}}}$, respectively, and $n(\mathcal{I}_{\bar{e}})$ is the number of slices used by solution $\mathcal{I}_{\bar{e}}$, then $\mathcal{A}_{\bar{e}}$ can be defined as:

$$\mathcal{A}_{\bar{e}} = \underset{\mathcal{I}_{\bar{e}} \in \mathcal{M}(\mathcal{P}^k_{\bar{e}}) \times \mathcal{M}(\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}) \times \mathcal{M}(\mathcal{S}_{\mathbb{P}^k_{\bar{e}}})}{\arg\min} n(\mathcal{I}_{\bar{e}}) \tag{12}$$

Our proposed algorithm for a single VLink embedding with already mapped endpoints (Algorithm 1) enumerates all such tuples in a brute-force manner and selects the one that uses the minimum number of spectrum slices. To accomplish this, we first enumerate all possible multi-sets $\mathbb{P}^k_{\bar{e}}$ of sizes ranging from 1 to q, *i.e.*, all possible selection of up to q paths from $\mathcal{P}^k_{\bar{e}}$. For each enumerated $\mathbb{P}^k_{\bar{e}} \in \mathcal{M}(\mathcal{P}^k_{\bar{e}})$, Algorithm 1 needs to compute $\mathbb{T}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ and $\mathbb{S}^{opt}_{\mathbb{P}^k_{\bar{e}}}$, the multi-sets of transmission configuration and slice indices, respectively, which minimizes $n(\mathbb{P}^k_{\bar{e}})$, the number of spectrum slices required for $\mathbb{P}^k_{\bar{e}}$. However, while computing $\mathbb{T}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ and $\mathbb{S}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ Algorithm 1 avoids enumerating all possible multi-sets by using the following technique.

Algorithm 1: Find solutions for a single VLink

```
1  function FindSolutions($G, \bar{e}, \mathcal{P}^k_{\bar{e}}, \mathcal{T}_{\bar{e}}$)
2      $\mathcal{M}(\mathcal{P}^k_{\bar{e}}) \leftarrow \phi$, $\mathtt{L}_{\bar{e}} \leftarrow \phi$
3      for $i \in N_q$ do
4          $\mathcal{M}(\mathcal{P}^k_{\bar{e}}) \leftarrow \mathcal{M}(\mathbb{P}^k_{\bar{e}}) \cup$ Multi-Set($\mathcal{P}^k_{\bar{e}}, i$)
5      foreach $\mathbb{P}^k_{\bar{e}} \in \mathcal{M}(\mathcal{P}^k_{\bar{e}})$ do
6          $\mathcal{M}(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}) \leftarrow$ Multi-Set($\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}, |\mathbb{P}^k_{\bar{e}}|$) s.t.
            $\sum_{d \in \mathbb{D}_{\mathbb{P}^k_{\bar{e}}}} d = \bar{\beta}_{\bar{e}}$
7          foreach $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}})$ do
8              $\zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}) \leftarrow$ AllPerm($\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$)
9              foreach $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ do
10                 $(n, \mathbb{P}, \mathbb{T}, \mathbb{S}) \leftarrow$ MDP($\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$)
11         Find $\mathbb{P}^{opt}$, $\mathbb{T}^{opt}$ and $\mathbb{S}^{opt}$ that minimizes
            $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ using (13)
12         $\mathcal{I}_{\bar{e}} \leftarrow (\mathbb{P}^{opt}, \mathbb{T}^{opt}, \mathbb{S}^{opt})$
13         $\mathtt{L}_{\bar{e}} \leftarrow \mathtt{L}_{\bar{e}} \cup \mathcal{I}_{\bar{e}}$
14     Sort $\mathtt{L}_{\bar{e}}$ in increasing order of $n(\mathcal{I}_{\bar{e}})$ in $\mathtt{L}_{\bar{e}}$
15     return $\mathtt{L}_{\bar{e}}$
```

For a path selection $\mathbb{P}^k_{\bar{e}}$, Algorithm 1 uses the reach table to enumerate the possible combination of $|\mathbb{P}^k_{\bar{e}}|$ data-rates such that their sum equals the VLink's demand $\beta_{\bar{e}}$. These combination of data-rates is represented by a multi-set $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} = (\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}, m_4)$, where $\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}$ is the set of all feasible data-rates for the paths in $\mathbb{P}^k_{\bar{e}}$ according to the reach table, and $m_4 : \mathcal{D}_{\mathbb{P}^k_{\bar{e}}} \rightarrow N_q$ defines the number of times a data-rate appears in $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$. Note that $|\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}| = |\mathbb{P}^k_{\bar{e}}|$ and $\sum_{d \in \mathbb{D}_{\mathbb{P}^k_{\bar{e}}}} d = \bar{\beta}_{\bar{e}}$.

Provisioning the same data-rate over different paths require different number of spectrum slices since the paths differ in both physical distance and hop count. To take this into account, for each $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$, Algorithm 1 considers all permutations of data-rates from $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$ (denoted by $\zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$) assigned to the paths in $\mathbb{P}^k_{\bar{e}}$ and computes the number of required spectrum slices for each assignment using (13) (lines 8 – 11).

$$n(\mathbb{P}^k_{\bar{e}}) = \min_{\forall \mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}) | \forall \mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}})} n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}) \tag{13}$$

Here, $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ is the number of slices required for a $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$, and $\mathcal{M}(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}})$ represents all possible multi-sets of $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$. $\mathbb{T}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ and $\mathbb{S}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ are then the transmission configuration selection and spectrum allocation to provision the data-rates to path assignment, requiring the minimum number of spectrum slices.

Algorithm 2: Algorithm for Computing $(n, \mathbb{P}, \mathbb{T}, \mathbb{S})$, given a multi-set $\mathbb{P}^k_{\bar{e}}$ and a permutation $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$

```
1  function MDP($\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$)
2      if $F[\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}] \neq \phi$ then
3          return $F[\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}]$
4      if $|\mathbb{P}^k_{\bar{e}}| = 1$ then
5          Find $t$ that minimizes $n_t$ for $p_1 \in \mathbb{P}^k_{\bar{e}}$ and
            $d_1 \in \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$ using (15)
6          if $t = \phi$ or First-Fit($p_1, n_t$) $= \phi$ then
7              $n \leftarrow \infty$, $\mathbb{P} \leftarrow \mathbb{T} \leftarrow \mathbb{S} \leftarrow \phi$
8          else
9              $n \leftarrow n_t$, $\mathbb{P} \leftarrow p_1$
10             $\mathbb{T} \leftarrow \{t\}$, $\mathbb{S} \leftarrow$ First-Fit($p_1, n_t$)
11         Store $(n, \mathbb{P}, \mathbb{T}, \mathbb{S})$ to $F$ and return
12     else
13         for $i = 1$ to $|\mathbb{P}^k_{\bar{e}}|$ do
14             $(n_i, \mathbb{P}_i, \mathbb{T}_i, \mathbb{S}_i) \leftarrow$ MDP($\{p_i\}, \{d_i\}$)
15         $n \leftarrow \sum_{i=1}^{|\mathbb{P}^k_{\bar{e}}|} n_i$, $\mathbb{P} \leftarrow \bigcup_{i=1}^{|\mathbb{P}^k_{\bar{e}}|} \mathbb{P}_i$
16         $\mathbb{T} \leftarrow \bigcup_{i=1}^{|\mathbb{P}^k_{\bar{e}}|} \mathbb{T}_i$, $\mathbb{S} \leftarrow \bigcup_{i=1}^{|\mathbb{P}^k_{\bar{e}}|} \mathbb{S}_i$
17         if $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}) = \infty$ then
18             Store $(\infty, \phi, \phi, \phi)$ to $F$ and return
19         else if All paths in $\mathbb{P}^k_{\bar{e}}$ are disjoint then
20             Store $(n, \mathbb{P}, \mathbb{T}, \mathbb{S})$ to $F$ and return
21         else
22             $\zeta(\mathbb{P}^k_{\bar{e}}) \leftarrow$ AllPerm($\mathbb{P}^k_{\bar{e}}$)
23             foreach $\mathrm{P}^k_{\bar{e}} \in \zeta(\mathbb{P}^k_{\bar{e}})$ do
24                 $\mathbb{S}_{new} \leftarrow \phi$, $\mathbb{P} \leftarrow$ PermToMultiSet($\mathrm{P}^k_{\bar{e}}$)
25                 foreach $p_i \in \mathrm{P}^k_{\bar{e}}$ do
26                     $\mathbb{S}_{new} \leftarrow \mathbb{S}_{new} \bigcup$
                        First-Fit($p_i, n(\{p_i\}, \{t_i^{(d)}\})$)
27                 Store $(n, \mathbb{P}, \mathbb{T}, \mathbb{S}_{new})$ to $F$ and return
28             if There is no valid slice assignment then
29                 Store $(\infty, \phi, \phi, \phi)$ to $F$ and return
```

A key challenge here is to efficiently evaluate (13) without computing for all possibilities. We observe that optimal computation of $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ exhibits optimal substructures and overlapping sub-problems, which allow us to leverage memoized Dynamic Programming (DP) to efficiently evaluate (13). Algorithm 1 invokes a procedure called Memoized Dynamic Programming (MDP) (Algorithm 2), which uses DP to compute $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ for a data-rate permutation $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ assigned to a path selection $\mathbb{P}^k_{\bar{e}}$ as follows.

Given $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} = \langle d_1, d_2, .., d_{|\mathbb{P}^k_{\bar{e}}|} \rangle$, an ordered set representing a permutation of $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$, Algorithm 2 divides the problem into $|\mathbb{P}^k_{\bar{e}}|$ sub-problems. Each of these sub-problems is represented as $(\{p_i\}, \{d_i\})$, where d_i is the i-th datarate in $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$ and p_i is the i-th path in $\mathbb{P}^k_{\bar{e}}$ (Line - 14). Each $d_i \in \mathrm{D}_{\mathbb{P}^k_{\bar{e}}} | m_4(d_i) > 1$ may result in multiple occurrences of the same sub-problems for paths $p_i \in \mathbb{P}^k_{\bar{e}}$ with $m_1(p_i) > 1$ (*i.e.*, overlapping sub-problems). These sub-problems also have optimal substructure property since the optimal solution for $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ can be computed using the following recurrence:

$$n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}) = \sum_{i=1}^{|\mathbb{P}^k_{\bar{e}}|} n(\{p_i\}, \{d_i\}) \tag{14}$$

These two properties allow us to memoize results from already solved sub-problems (line 28) and use them for subsequent recursive invocations of Algorithm 2 (lines 2 – 3). Algorithm 2 uses (15) to find the most spectrally efficient transmission configuration t such that $t^{(d)} = d_i$ and t results in a transmission reach of at least the length of p_i.

$$n(\{p_i\}, \{d_i\}) = \begin{cases} \min_{\forall t \in \mathcal{T}_{\bar{e}p_i} | t^{(d)} = d_i \wedge r_t \geq l(p_i)} n_t \\ \infty \text{ if no } t \text{ can be found} \end{cases} \quad (15)$$

In addition to $n(\{p_i\}, \{d_i\})$, Algorithm 2 also returns the corresponding transmission configuration and a slice assignment consisting of contiguous and free n_t slices for p_i. Currently, we are using a First-fit approach to find slice assignment [7]. Note that it is possible that the SLinks on SPath p_i do not have n_t contiguous slices, rendering the sub-problem infeasible to solve (lines 17 – 18 and 29 – 30).

Algorithm 2 combines the transmission configurations returned by the sub-problems to obtain a valid transmission configuration selection $\mathbb{T}$ for the original problem. However, combining slice index selections $\mathbb{S}$ requires extra care for the following reasons. First, spectrum resources are shared among the SPaths in $\mathbb{P}^k_{\bar{e}}$ with common SLinks. Second, sub-problems are solved independently without considering the shared spectrum resources on the common SLinks between the SPaths. As a result, slice index selection of two sub-problems containing two paths with a common SLink can end up using the same spectrum slice. Therefore, these slice selections may become invalid when merging the solutions for the sub-problems for violating the requirement for non-overlapping spectrum. This also holds for the solution to the sub-problems involving the same paths. However, for the paths in $\mathbb{P}^k_{\bar{e}}$ that are disjoint, their slice index assignments from the sub-problems can be readily combined (line 19 – 20).

For the paths in $\mathbb{P}^k_{\bar{e}}$ that are not disjoint, Algorithm 2 has to select $n(\{p_i\}, \{t_i^{(d)}\})$ contiguous and un-occupied slices along each SPath $p_i \in \mathbb{P}^k_{\bar{e}}$. The order in which Algorithm 2 assigns slices to these paths has an impact on the subsequent slice selection. For instance, slice selections to $|\mathbb{P}^k_{\bar{e}}| - 1$ paths may fragment the spectrum on one of the SPaths in $\mathbb{P}^k_{\bar{e}}$ in such a way that the SPath does not have the required number of contiguous and un-occupied slices for the final slice assignment. To circumvent this issue, Algorithm 2 enumerates all permutations of the SPaths in $\mathbb{P}^k_{\bar{e}}$ (denoted by $\zeta(\mathbb{P}^k_{\bar{e}})$) to find a feasible slice selection, $\mathbb{S}_{new}$. For a given permutation $\mathrm{P}^k_{\bar{e}} \in \zeta(\mathbb{P}^k_{\bar{e}})$, it selects first available $n(\{p_i\}, \{t_i^{(d)}\})$ contiguous slices on each path $p_i \in \mathrm{P}^k_{\bar{e}}$ in the order p_i appears in $\mathrm{P}^k_{\bar{e}}$. However, Algorithm 2 may not find $\mathbb{S}_{new}$ even after exploring all permutations in $\zeta(\mathbb{P}^k_{\bar{e}})$ resulting in infeasible solution.

1) Running Time Analysis: Algorithm 1 explores all the possibilities of choosing the multi-set $\mathbb{P}^k_{\bar{e}}$. The number of multi-sets of cardinality q on ground set of k SPaths is given by $\left(\!\binom{k}{q}\!\right) = \binom{k+q-1}{q}$ [20]. Since $|\mathbb{P}^k_{\bar{e}}|$ can take any integer value from 1 to q, we get $|\mathcal{M}(\mathcal{P}^k_{\bar{e}})| = \sum_{i=1}^{q} \left(\!\binom{k}{i}\!\right)$. For each $\mathbb{P}^k_{\bar{e}}$, Algorithm 1 enumerates all possible multisets $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}})$, resulting in $\binom{|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|+q-1}{q}$ enumerations. Algorithm 1 invokes Algorithm 2 for all the permutations of each of the multisets $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}})$. The number of permutations of a multiset $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$ of cardinality q is given by the multinomial coefficient, *i.e.*, $\frac{q!}{\Pi_{d_j \in \mathcal{D}_{\mathbb{P}^k_{\bar{e}}}} m_4(d_j)!}$ [21]. Therefore, Algorithm 1 invokes Algorithm 2 $\frac{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|+q-1)!}{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|-1)! \times \Pi_{d_j \in \mathcal{D}_{\mathbb{P}^k_{\bar{e}}}} m_4(d_j)!}$ times to compute $n(\mathbb{P}^k_{\bar{e}})$. The most expensive step of Algorithm 2 is the exploration of all the permutations of the paths in $\mathbb{P}^k_{\bar{e}}$ requiring $\frac{q!}{\Pi_{p_j \in \mathcal{P}^k_{\bar{e}}} m_1(p_j)!}$ possibilities in the worst case. Therefore, to find $\mathcal{A}_{\bar{e}}$, Algorithm 1 enumerates $\left(\sum_{i=1}^{q} \binom{k+i-1}{i} \right) \times \frac{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}| + q - 1)!}{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}| - 1)! \times \Pi_{d_j \in \mathcal{D}_{\mathbb{P}^k_{\bar{e}}}} m_4(d_j)!} \times \frac{q!}{\Pi_{p_j \in \mathcal{P}^k_{\bar{e}}} m_1(p_j)!}$ possibilities. Typical values of k and q are small, therefore, the running time is dominated by the size of $\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}$.

B. Heuristic solution for VN Embedding

Algorithm 3: Algorithm for VN Embedding

```
1  function VNEmbedding($G, \bar{G}$)
2      Perform VNode mapping to compute $\tau : \bar{V}$
3      $\bar{E}_{sort} \leftarrow$ Sort all $\bar{e} \in \bar{E}$ in descending order of $\bar{\beta}_{\bar{e}}$
4      foreach $\bar{e} \in \bar{E}_{sort}$ do
5          $\mathtt{L}_{\bar{e}} \leftarrow$FindSolutions($G, \bar{e}, \mathcal{P}^k_{\bar{e}}, \mathcal{T}_{\bar{e}}$)
6          foreach $\mathcal{I}_{\bar{e}} \in \mathtt{L}_{\bar{e}}$ in sorted order do
7              if $\exists \bar{e} \in \bar{E}_{\bar{e}r}$ will be blocked by $\mathcal{I}_{\bar{e}}.\mathbb{S}$ then
                 continue
8              foreach $e \in p | p \in \mathcal{I}_{\bar{e}}.\mathbb{P}$ do
9                  Perform slice assignment using $\mathcal{I}_{\bar{e}}.\mathbb{S}$
10             $\chi_{\bar{e}}.\mathbb{P} \leftarrow \mathcal{I}_{\bar{e}}.\mathbb{P}$, $\chi_{\bar{e}}.\mathbb{T} \leftarrow \mathcal{I}_{\bar{e}}.\mathbb{T}_{\mathbb{P}^k_{\bar{e}}}$
11             break
12         if $\chi_{\bar{e}} = \phi$ then
13             return $(\phi, \phi)$
14     return $(\tau : \bar{V} \rightarrow V, \gamma : \bar{E} \rightarrow \chi)$
```

We present a two-stage algorithm for computing embedding of a VN $\bar{G}$ on an EON G in Algorithm 3. The first stage computes VNode embedding by mapping each VNode to an SNode in its location constraint set in a random manner. The second stage computes VLink embedding based on the output of the first stage. Indeed, a sequential VNode and VLink embedding constraints the solution space. However, in this paper our main focus is on addressing the complexities associated with obtaining efficient VLink embedding. We leave the joint VNode and VLink embedding as a future work.

Computing the optimal VLink mapping with a given VNode mapping involves considering all $|\bar{E}|!$ VLink ordering and for each ordering sequentially embedding the VLinks by invoking Algorithm 1. The lowest cost mapping among all possible VLink ordering will be the optimal solution for the given VNode mapping. However, this brute-force approach does not scale with the number of VLinks. Instead, Algorithm 3 considers only one VLink ordering to find a solution within a reasonable time. It employs a look-ahead technique for reducing blocking probability. First, it sorts the VLinks in the descending order of their demands (line 3) since a VLink with lower demand is more likely to cause spectrum fragmentation. For each VLink $\bar{e}$ in the sorted order, Algorithm 3 computes the feasible solutions sorted in increasing order of spectrum occupancy by invoking Algorithm 1 (line 5). Algorithm 3 iterates over the solutions in increasing order of their spectrum occupancy and stops at the first solution, which if chosen does not block the remaining VLinks in $\bar{E}$ denoted as $\bar{E}_{\bar{e}r}$. This results in selecting the minimum cost solution that does not block the embedding of the VLinks in $\bar{E}_{\bar{e}r}$ (lines 6 – 11). Algorithm 3 allocates spectrum slices on all SLinks present in that solution and updates the VLink mapping function χ. If no solution for $\bar{e}$ can be found, the VN embedding is rejected.

VI. Evaluation

A. Simulation Setup

We implement the ILP formulation presented in Section IV using IBM ILOG CPLEX C++ libraries and compare it with a C++ implementation of the heuristic presented in Section V. We use Nobel Germany (17 nodes and 26 links) and Germany50 (50 nodes and 88 links) networks from SNDlib repository [13] as the SNs for small and large scale simulations, respectively. Each SLink is considered to have two optical fibers in opposite directions, each with spectrum bandwidth of 600GHz and 4THz for small and large scale simulations, respectively. We generate the VNs by varying link to node ratio (LNR), while keeping number of VNodes fixed (*i.e.*, 8 and 50 for small and large scale simulations, respectively). A VN with a higher LNR has a higher density and total bandwidth demand than a VN with lower LNR. For each VN, location constraints of VNodes are chosen randomly, and VLink demands are varied between 100Gbps to 1Tbps with possible values as multiples of 100Gbps. $k = 10$ shortest paths between all pairs of SNodes in each SN are pre-computed as input to our simulation. We simulate the embedding of those VNs independently on the corresponding SN for each of the compared approaches discussed in the next subsection. In each simulation run, we generate 5 and 50 different VNs with the same LNR with similar total bandwidth demands for small and large scale simulations, respectively, and take the mean of each performance metric over those VNs. Simulations are performed on a machine with 8×10-core Intel Xeon E7-8870 2.40GHz processors and 1TB of RAM.

B. Compared Variants

In order to quantify the impact of different degrees of freedom (DoF) in an EON such as flexible spectrum allocation, variable modulation, and FEC, we compare the variants shown in Table I. Each DoF produces two variants in the table based on the binary choice of leveraging a DoF or not, resulting in total 8 variants. For instance, by allowing or not allowing flexible spectrum allocation while only one modulation format and FEC for each data rate in the reach table $\mathcal{R}$, we get the variants: *Fixed-FM-FF* and Flex-FM-FF. They differ in terms of spectrum slice allocation granularity (*i.e.*, 50GHz and 12.5GHz for fixed-grid and flex-grid, respectively) and the total number of possible data rates (*i.e.*, 100G, 200G, 400G for fixed-grid and 100G, 150G, 200G, 250G, 300G, 400G, 500G, 600G, and 800G for flex-grid). The spectrum slice width and the possible data rates for each variant are chosen based on current industry standards [22]. For both fixed- and flex-grid, variable modulation and/or FEC allow(s) us to choose from a variety of modulation formats and/or FECs for each data rate in $\mathcal{R}$, resulting in different reaches. The most flexible variant in Table I is Flex-VM-VF, which offers a variety of modulation formats and FECs to be used for each data rate, while at the same time enabling flexible spectrum allocation.

TABLE I
Compared Variants

Degrees of freedom		**Fixed-grid**	**Flex-grid**
Fixed FEC	Fixed Modulation	Fixed-FM-FF	Flex-FM-FF
	Variable Modulation	Fixed-VM-FF	Flex-VM-FF
Variable FEC	Fixed Modulation	Fixed-FM-VF	Flex-FM-VF
	Variable Modulation	Fixed-VM-VF	Flex-VM-VF

C. Performance Metrics

1) *Percentage of spectrum usage (PSU):* The total number of spectrum slices required to embed a VN computed using (1) and normalized by the total number of slices in the SN.

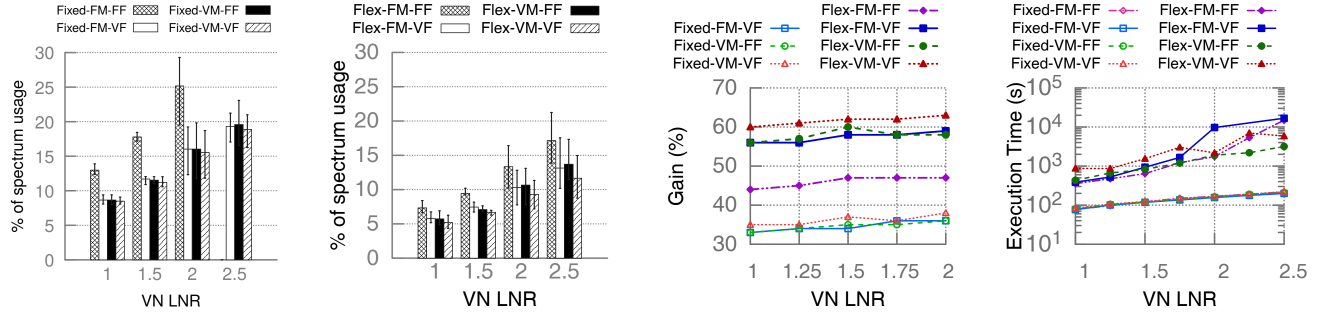
2) *Execution Time (ET):* The time required for an algorithm to find an embedding of a VN.

3) *Gain of X over Y:* The percentage spectrum savings of using variant X instead of variant Y.

D. Results and discussion

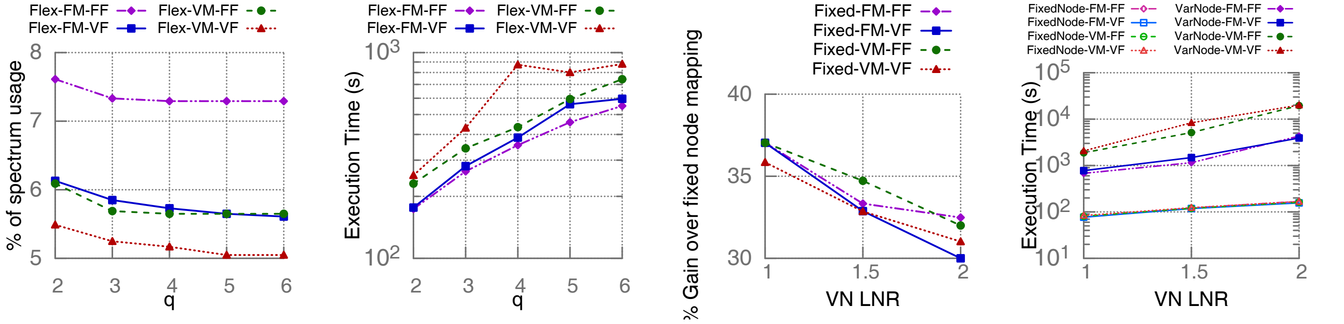
1) *Small-Scale Scenario (Nobel Germany):* In Fig. 1, we show the results of the ILP formulation to demonstrate the impact of different DoFs on our performance metrics. To accurately quantify these impacts, we use the same node mapping (randomly chosen) and fix the value of the maximum number of splits $q = 4$ for all the compared variants. Fig. 1(a) and Fig. 1(b) show the resulting PSU, for increasing value of LNR, in the case of fixed- and flex-Grid, respectively, for different variants. Fig. 1(a) and Fig. 1(b) confirm that adding more DoFs on top of the base cases (*Fixed-FM-FF* and *Flex-FM-FF*, respectively) enables significant spectrum gains. Less intuitively, the gains obtained by introducing variable modulation formats are very similar to the gains obtained by introducing FEC variability (typically these savings are slightly higher for low LNR and slightly lower for high LNR). We remark that these results are obtained using a reach table listing a large number of transmission configurations (on average 4 options for modulation formats and 3 for FEC). Having several options for modulation or FEC seem to be mostly equivalent, with a little advantage depending on the LNR. Moreover, combining the two DoFs (*i.e.*, *Fixed-VM-VF* and *Flex-VM-VF* cases) provides an additional gain only for large LNRs, demonstrating again that, if a sufficient number of alternative modulation (or FEC) options are available, significant gains can be achieved using only one DoF.

Fig. 1(c) helps us visualizing the percentage gains of all the variants with respect to the baseline case *Fixed-FM-FF*. Gains span from about 30% in the case of DoFs added over the fixed grid, to about 45% when introducing the flex-grid, up to 60% when additionally introducing DoFs in the flex-grid. Another interesting trend arising in these figures is that



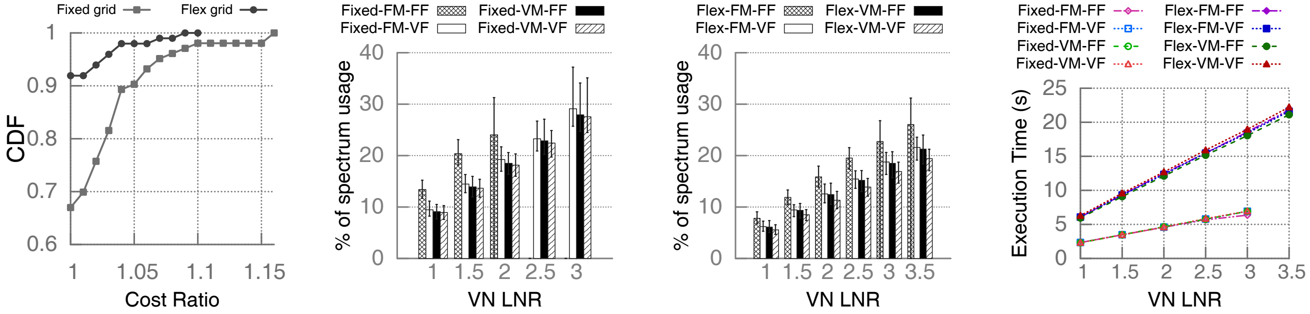
(a) Fixed grid (b) Flex grid (c) Fixed vs Flex Grid (d) Execution Time

Fig. 1. Spectrum Usage and Execution Time Analysis for Small Scale Topologies



(a) Impact of number of splits (b) Impact of number of splits (c) Impact of variable node mapping (d) Impact of variable node mapping

Fig. 2. Impact of VLink Splitting and Variable VNode Mapping



(a) Optimality of the Heuristic (b) Fixed grid (c) Flex grid (d) Execution Time

Fig. 3. Spectrum Usage and Execution Time Analysis for Heuristic

both PSU and percentage gains monotonically increase for increased LNR. The reason is twofold. First, a higher number of VLinks introduces more chances of optimization, justifying the increasing trend of gains. Second, a higher LNR induces a more constrained scenario as a higher stress is put on the SLinks incident to the SNode where a VNode is mapped. The effect of such a constrained scenario is obvious from Fig. 1(a) where it can be observed that *Fixed-FM-FF* becomes infeasible for an LNR=2.5, even though spectrum utilization is not significantly high. Finally, Fig. 1(d) shows the ETs of the ILP for the 8 problem variants. Following the fact that flex-grid variants have 3× more data rates and 4× more slices compared to fixed-grid variants, flex-grid variants are on average two orders of magnitude slower than the fixed-grid variants. More interestingly, the impact of DoFs on ET is negligible in the case of fixed-grid, and very limited in the case of flex-grid.

To investigate the impact of the maximum number of splits q, we plot PSU and ET with LNR=1.0 using flex-grid variants in Fig. 2(a) and Fig. 2(b), respectively. We omit the case where $q = 1$ as no transmission configuration with the latest hardware can serve the VLink demand 1Tbps without any split. However, with $q = 2$, 1Tbps demands can be served by two splits of spectrum slices, each providing 500Gbps, which is realizable for shorter distances. Fig. 2(a) shows that the marginal gain of increasing q from 2 to 4 tends to decrease, and after $q = 4$, PSU converges to a steady value. This is due to the fact that, after a certain number of splits, the effect of volume discount becomes prominent and discourages further splitting. On the other hand, higher values of q have drastic impact on the complexity of the ILP formulation as shown by the higher ET for a higher q in Fig. 2(b). Based on these results, we set $q = 4$ for the other simulations in this paper.

Finally, Fig. 2(c) and Fig. 2(c) evaluate the impact of considering multiple SNodes as the candidates for a VNode compared to the fixed node mapping case. Fig. 2(c) shows that more than 30% spectrum can be saved by considering node mapping variability as opposed to fixed node mapping. This is due to the fact that node mapping variability allows to map VNodes to the most optimized SNodes in terms of spectrum usgae. As VN LNR increases the options for optimizing diminishes, resulting in the downward trend with increasing LNR. Fig. 2(c) shows that variants with variable node mapping are an order of magnitude slower than fixed node mapping counterparts. This is due to the increased complexity introduced by constraint (10). In addition, (10) magnifies the intricacy of flex-grid variants turning them impossible to solve within bounded times with our current simulation settings.

We focus now on the comparison of solution quality of our heuristic approach benchmarking against the ILP results on the small-scale network. Fig. 3(a) plots the Cumulative Distribution Function (CDF) of the cost ratios between the heuristic algorithm and the ILP formulation (we merge results for all the fixed- and flex-grid variants, respectively). We perform VNE with our heuristic algorithm for the same set of VNs in which ILP formulation was able to find solutions (i.e, the solution shown in Fig. 1). It is worth to mention that the heuristic was able to find solutions for 88% and 97% of the VNs for fixed- and flex-grid variants, respectively. Among the fixed-grid variants, *Fixed-FM-FF* has the highest percentage (*i.e.*, 34%) of not finding a solution due to its inferior resource efficiency. The main reason for the heuristic failing to find a solution is the fact that, to limit time complexity, it enumerates only one permutation of VLinks in the VN in a greedy manner. This fact, combined with the First-fit spectrum assignment, create spectrum fragmentation in SLinks and might induce blocking of later VLinks. However, when the heuristic finds a solution it remains very close to the optimal solution generated by the ILP formulation: 98th percentile of VNs have a cost ratio less than 9% and 5% for fixed- and flex-grid, respectively. The heuristic tends to perform better for flex-grid variants as they have more alternative solutions close to the optimum that can be more easily found by the heuristic.

2) Large-Scale Scenario (Germany50): Given the small optimality gap of the heuristic, we now use the heuristic to show the impact of different DoFs in large scale networks in Fig. 3(b) and Fig. 3(c). Results on this larger topology confirm the conclusions observed in Fig. 1(a) and Fig. 1(b) earlier, but we remark that now the same conclusions are obtained on more dense VNs and by repeating our simulations over a much higher number of instances to achieve statistically confident results. Finally, Fig. 3(d) shows that the execution time of our heuristic grows linearly with LNR and with the addition of more flexibility conforming to the analysis in V-A1.

VII. Conclusion

In this paper, we proposed an optimization model that captures the full flexibility in all the transmission parameters of an EON, namely flexible spectrum allocation, baud rate selection, adaptive modulation format and FEC overhead, while considering virtual link splitting to solve the VNE over EON problem. We also devised a heuristic that obtains near optimal solutions while executing several orders of magnitude faster than the optimal solution. Extensive simulations using real network topologies showed that considering flexibility in all transmission parameters can lead to significant savings in spectrum resources (up to 60% in our evaluation) compared to the case when no flexibility is considered. Simulation results also showed that heuristic solutions remain within 5% of the optimal solutions in more than 90% of the cases.

We hope that these promising results will further stimulate research in achieving a fully flexible EON. Interesting research directions include recovery of VNs after substrate failures and re-optimization of embedding by leveraging EON flexibilities.

Acknowledgement

This work was funded in part by Huawei Canada and in part by an NSERC CRD Grant. This work benefited from the use of CrySP RIPPLE Facility at University of Waterloo.

References

[1] L. Peterson *et al.*, "Central office re-architected as a data center," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 96–101, 2016.

[2] "Cisco vni forecast highlights." [Online]. Available: https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html

[3] O. Gerstel *et al.*, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Communications Magazine*, vol. 50, no. 2, 2012.

[4] A. Pagès *et al.*, "Optimal route, spectrum, and modulation level assignment in split-spectrum-enabled dynamic elastic optical networks," *Journal of Optical Comm. and Net.*, vol. 6, no. 2, pp. 114–126, 2014.

[5] R. Boutaba *et al.*, "Elastic optical networking for 5g transport," *Journal of Network and Systems Management*, vol. 25, no. 4, pp. 819–847, 2017.

[6] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[7] B. C. Chatterjee *et al.*, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 3, pp. 1776–1800, 2015.

[8] N. Sambo *et al.*, "Routing, code, and spectrum assignment, subcarrier spacing, and filter configuration in elastic optical networks," *Journal of Optical Comm. and Net.*, vol. 7, no. 11, pp. B93–B100, 2015.

[9] A. Fischer *et al.*, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 1888–1906, Feb 2013.

[10] Y. Wang *et al.*, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1503–1511.

[11] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *Journal of Lightwave Technology*, vol. 32, no. 3, pp. 450–460, 2014.

[12] R. Lin *et al.*, "Virtual network embedding with adaptive modulation in flexi-grid networks," *Journal of Lightwave Technology*, 2017.

[13] Sndlib repository [online] http://sndlib.zib.de/home.action.

[14] J. Zhao *et al.*, "Virtual topology mapping in elastic optical networks," in *Proc. of IEEE ICC*, 2013, pp. 3904–3908.

[15] M. Jinno *et al.*, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]," *IEEE Communications Magazine*, vol. 48, no. 8, 2010.

[16] C. Rottondi *et al.*, "Routing, modulation format, baud rate and spectrum allocation in optical metro rings with flexible grid and few-mode transmission," vol. 35, no. 1, pp. 61–70, 2017.

[17] N. Sambo *et al.*, "Routing code and spectrum assignment (RCSA) in elastic optical networks," *Journal of Lightwave Technology*, vol. 33, no. 24, pp. 5114–5121, 2015.

[18] D. Garrido *et al.*, "Routing, code, modulation level and spectrum assignment (RCMLSA) algorithm for elastic optical networks," in *Proc. of IEEE ICTON*, 2016, pp. 1–4.

[19] N. Amaya *et al.*, "Introducing node architecture flexibility for elastic optical networks," *Journal of Optical Communications and Networking*, vol. 5, no. 6, pp. 593–608, 2013.

[20] S. Pemmaraju and S. Skiena, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica®*. Cambridge university press, 2003.

[21] R. A. Brualdi, "Introductory combinatorics," *New York*, vol. 3, 1992.

[22] Itu-t g.694.1. spectral grids for wdm applications: Dwdm frequency grid.[online https://www.itu.int/rec/t-rec-g.694.1/en.