Virtual network mapping in elastic optical networks with sliceable transponders

Juzi Zhao & Suresh Subramaniam

ONLINE FIRST

Photonic Network Communications

ISSN 1387-974X

Photon Netw Commun
DOI 10.1007/s11107-020-00892-x

Photonic Network Communications
https://doi.org/10.1007/s11107-020-00892-x

ORIGINAL PAPER

Virtual network mapping in elastic optical networks with sliceable transponders

Juzi Zhao[1] · Suresh Subramaniam[2]

Received: 11 November 2019 / Accepted: 17 June 2020

Abstract
Emerging inter-datacenter applications involving data transferred, processed, and analyzed at multiple datacenters, such as virtual machine migrations, real-time data backup, remote desktop, and virtual datacenters, can be modeled as virtual network requests that share computing and spectrum resources of a common substrate physical inter-datacenter network. Recent advances make elastic optical networks an ideal candidate for meeting the dynamic and heterogeneous connection demands between datacenters. In this paper, we address the static (offline) version of the virtual network mapping problem in elastic optical networks equipped with sliceable bandwidth variable transponders. The objective is to minimize the total transponder cost in the network. An Integer Linear Programming formulation is presented, lower bounds are derived, and six heuristics are proposed and compared. Simulation results are presented to show the effectiveness of the proposed approaches.

Keywords Virtual networks · Elastic optical networks · Inter-datacenter networks · Frequency slots · List scheduling

1 Introduction

Elastic optical networks (EONs), with a fine-grained spectral spacing and variable channel center frequencies, are promising for connecting geo-distributed datacenters due to their capability to transfer large amounts of data as well as efficiency and flexibility of allocating spectrum resources to heterogeneous traffic demands [1, 2].

Network virtualization breaks the network's rigidly fixed pattern by enabling different virtual networks with various requirements to share the same substrate physical network. Each virtual network (VN) request is a logical topology including a set of virtual nodes interconnected by virtual links. These requests require not only communication resources on the optical links but also compute and storage resources at the physical nodes in the network. Emerging inter-datacenter applications, such as virtual machine migration, database replication/backup, and virtual datacenters, can be modeled as VN requests. The problem of allocating route(s) as well as appropriate spectrum, computing, and storage resources to a request or set of requests is commonly known as the virtual network mapping/embedding problem (VNM/VNE), which is proved to be NP-hard in [3].

There are several papers on VNM/VNE in EONs. The authors of [4] consider both static and dynamic cases, with the objective of maximizing spectrum slot utilization in static case and minimizing blocking ratio in dynamic case. Traffic grooming is taken into account in the problem of network virtualization in [5] over both WDM networks and EONs. In [6], an ILP formulation and heuristics to minimize the total spectrum slot usage are proposed for both transparent and opaque VNs. Two ILP formulations for VNM problem are presented in [7]. The authors of [8] take fragmentation-awareness and load-balancing features into account for the VNM, with an ILP formulation proposed. The authors of [9] propose relaxed ILP and heuristics for static version of VNM based on re-optimized virtual concatenation (VCAT) framework. An ILP formulation to maximize the number of successful mappings over transparent EONs considering transmission distance limits is proposed in [10], but only small problem instances can be solved. The authors of [11] develop an ILP and a heuristic algorithm to minimize the total network cost considering impairment constraints and

✉ Juzi Zhao
juzi.zhao@sjsu.edu

Suresh Subramaniam
suresh@gwu.edu

[1] San Jose State University, One Washington Square, ENG371, San Jose, CA 95112, USA

[2] The George Washington University, Washington, DC, USA

Published online: 07 July 2020

single link failures. The authors of [12] construct an auxiliary graph and utilize spectrum partitioning approaches to solve the cost-effective survivable VNM problem. Energy efficient VNM schemes are proposed for dynamic transparent VN requests in [13] with the same spectrum bandwidth allocated to all virtual links of the VN request.

Sliceable bandwidth variable transponders (SBVTs, also called multi-flow transponders (MFT)) have been recently adopted in EONs. Multiple lightpaths with different destinations can be launched by a single SBVT, and thus transponder resources are effectively utilized. On the other hand, each normal bandwidth variable transponder (BVT) is only capable of launching a single lightpath.

Many research directions in SBVT-enabled EONs have been explored. The authors of [14] propose dynamic routing, spectrum, and transponder assignment schemes for two different types of SBVTs. The authors of [15] devise a dynamic distance-adaptive routing, spectrum, and modulation format assignment algorithm to optimize the utilization of spectrum and SBVTs. A model based on auxiliary graph is proposed in [16] to investigate the dynamic routing and spectrum assignment problem with both electrical and optical grooming.

In this paper, we consider the static version of a VNM problem in EONs with SBVT and BVT allocation that has not been addressed before, to the best of our knowledge. A set of VN requests is given, where each request consists of a set of virtual nodes and the amount of computation resources required by the virtual nodes and the bit rate required by virtual links connecting virtual nodes. The objective is to minimize the total transponder cost for all requests. We present an Integer Linear Programming (ILP) formulation, two lower bounds, and six heuristics based on list scheduling and Tabu search to map the VN requests onto the physical optical inter-datacenter network, and assign VMs, frequency slots, and transponders to the requests. The problem of static VNM problem in EONs with SBVTs was investigated for the first time in our previous work [17]. In this paper, we extend that work by proposing two new heuristics. In addition, in this paper, the physical network includes both SBVTs and BVTs as transponder resources.

The paper is organized as follows: In Sect. 2, we present the network model and the detailed problem statement. An ILP formulation, lower bounds, and heuristics are presented in Sect. 3. Section 4 presents and discusses the simulation results. We conclude the paper in Sect. 5.

2 Model and problem statement

The physical inter-datacenter network topology is $G_p(V_p, E_p)$, where V_p is the set of nodes (each is a datacenter associated with an optical switch node), and E_p is the set of links. Each physical node has H virtual machines (VMs). Some physical nodes are equipped with SBVTs, while the other physical nodes are equipped with BVTs. Each SBVT has C carriers. Each BVT has only one carrier. All the SBVTs and BVTs have the same capacity. The cost of each SBVT is Γ, and the cost of each BVT is Θ. There are two fibers on each physical link with opposite directions, and each has S frequency slots. The bandwidth of each frequency slot is 12.5 GHz. A number of frequency slots are allocated to each lightpath according to its spectrum requirement. The frequency slots allocated to each lightpath have to be contiguous (spectrum contiguity constraint), and the same band of frequency slots has to be allocated on each link traversed by the lightpath (spectrum continuity constraint). A guardband of G consecutive frequency slots has to be assigned between adjacent frequency slot bands allocated to different lightpaths if they share common physical links, in order to avoid interference [18]. K shortest distance paths between each pair of nodes are precomputed. One of M possible modulation formats is assigned to each lightpath. Each modulation format has a maximum transmission reach associated with it. A lightpath assigned modulation format m must have a path distance less than R_m. Depending on the path length, each precomputed path has a highest modulation level that can be used. The physical network is considered as transparent; therefore, no spectrum converters exist in the network.

A VN request i can be represented as a graph $G_l^i(V_l^i, E_l^i)$, where V_l^i is the set of virtual nodes, and E_l^i is the set of virtual links. For clarity, we will drop the subscript l which indicates a virtual or logical topology. There is a bit rate requirement Λ_k^i for each virtual link $k^i \in E^i$ for bidirectional communication between the two end virtual nodes. Each virtual node j^i can be mapped to one physical node of a candidate physical node set (N_j^i) and requires h_j^i VMs [6, 7, 9–13]. Please note that the case of fixed virtual node mapping could be treated as a special case, where the candidate set of each virtual node has only one physical node. According to the candidate physical node sets of two end virtual nodes, virtual link k^i has a candidate path set P_k^i. (For example, suppose a virtual link k with end virtual nodes v_1 and v_2, and virtual node v_1 has two candidate physical nodes n_1 and n_2, and virtual node v_2 has two candidate physical nodes n_3 and n_4, then the candidate path set for virtual link k includes the paths pairs (n_1, n_3), (n_1, n_4), (n_2, n_3), and (n_2, n_4).) Path $p \in P_k^i$ has a frequency slot requirement $b_k^i(p)$, based on the path's highest modulation format m^p and Λ_k^i, i.e., $b_k^i(p) = \lceil \Lambda_k^i/(s_p * 12.5) \rceil + G$ (including guardband), where s_p is the spectrum efficiency of m^p in Gbps/GHz. In this paper, we assume two virtual nodes of the same VN request can be mapped to one physical node as long as the VM resources permit. In this case, there are no frequency slots nor transponders allocated to the virtual link connecting

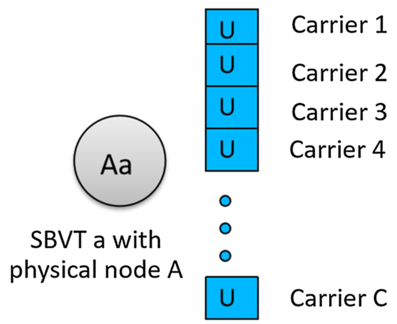

Fig. 1 SBVT model

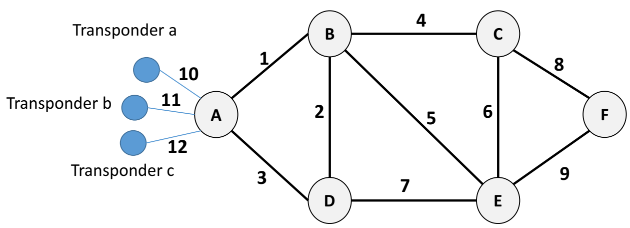

Fig. 2 Extra links introduced by transponders

these two virtual nodes. (We note that the proposed approaches can be easily modified if this assumption does not hold, by adding an additional constraint that no two virtual nodes of the same request can be mapped to a single physical node.)

In this paper, we adopt the multi-laser sliceable bandwidth variable transponder (SBVT) model [14]. According to this model, each SBVT has C carriers [16]. A dedicated tunable laser is associated with each carrier, and the bandwidth of each carrier is U GHz [16] as shown in Fig. 1. A single lightpath can be assigned 1 to C carriers based on its bandwidth requirement. For example, a lightpath that requires $3U$ GHz bandwidth needs three carriers. If more than one carrier is assigned to a single lightpath, a 2 GHz guardband will be allocated between two carriers due to the center-frequency instability issue with lasers [14]. Therefore, $3U + 4$ GHz is required by the lightpath with $3U$ GHz bandwidth requirement. Independent lightpaths (possibly for different virtual requests) can share the same SBVT as long as the carrier limit is not exceeded. No electrical processing is needed at intermediate nodes. Note that if multiple lightpaths are assigned the same SBVT at a particular node, then the bandwidths allocated to these lightpaths cannot have any overlap. This leads to the idea of extra links in the physical network as shown in Fig. 2. (Only three transponders at node A are shown in the figure for illustration purpose.) The original network has nine links (links 1–9). The links between transponders and physical nodes can be treated as extra links (links 10–12). All lightpaths assigned transponder a at node A will transverse extra link 10. (Please note that only one lightpath will be assigned transponder a if it is a BVT.) Recall that for EONs, a single frequency slot on each link can be assigned to at most one lightpath (spectrum clash constraint). This constraint applies to both physical links and extra links.

We consider the static version of the VNE problem where a set of requests is given *a priori*. Each virtual node in a request is mapped to a single physical node, with sufficient available VMs, selected from its candidate physical node set. Each virtual link is mapped to a physical path p, and a band of contiguous frequency slots is allocated to the virtual link based on its bit rate requirement. Two transponders (BVT or SBVT) on the source and destination nodes, respectively, of the path p have to be assigned to launch and terminate the lightpath, and a corresponding number of carriers on the assigned SBVTs are allocated to the virtual link. (If a BVT is allocated, the entire BVT with one carrier is allocated to the virtual link.) The objective is to minimize the total cost of transponders to map all requests according to the number of required SBVTs and BVTs, i.e., if n_1 SBVTs and n_2 BVTs are allocated, the total transponder cost is $n_1\Gamma + n_2\Theta$.

3 Algorithms

In this section, we present our approaches to address the VNM problem. We first present an ILP formulation that can provide an optimal solution, but it is computationally intensive and can only be applied to small problem instances. We then present two lower bounds for the problem, which can be used as a comparison reference for heuristic results for large problem instances. After that, we present six lower complexity heuristics.

3.1 ILP formulation

The input parameters and variables in ILP formulation are shown in Tables 1 and 2, respectively.

$$\text{Objective: Minimize } \sum_n \delta_n \left(\sum_q m_{nq} \right)$$

Constraints:

(1) $\sum_n x_{ijn} = 1$ for all i, j;
(2) $\sum_n a_{ijn} x_{ijn} = 1$ for all i, j;
(3) $\sum_{ij} h^i_j x_{ijn} \leq H_n$ for all n;
(4) $\sum_p y_{ikp} = 1$ for all i, k
(5) $\sum_p y_{ikp} S(p) = \sum n x_{iS(k^i)n}$ for all i, k
(6) $\sum_p y_{ikp} D(p) = \sum n x_{iD(k^i)n}$ for all i, k
(7) $\tau_{iks_0} = g_{iks_0}$ for all i, k;
(8) $\tau_{iks} \geq g_{iks} - g_{iks-1}$ for all i, k, s
(9) $\sum_s g_{iks} = \sum_p b^i_k(p) y_{ikp}$ for all i, k;

Table 1 Notation of input parameters

Symbol	Meaning
a_{ijn} (binary)	$a_{ijn} = 1$ if physical node n is in the candidate set (N_j^i) of physical nodes to map to for virtual node j in request i
h_j^i (integer)	The number of VMs required by virtual node j in request i
H_n (integer)	The number of VMs on physical node n
$S(p)$ (integer)	The source physical node of physical path p
$D(p)$ (integer)	The destination physical node of physical path p
$S(k^i)$ (integer)	The source virtual node of virtual link k in request i
$D(k^i)$ (integer)	The destination virtual node of virtual link k in request i
$b_k^i(p)$ (integer)	The number of frequency slots (including guardband) required by virtual link k in request i if it is mapped to physical path p
$c_k^i(p_s)$ (integer)	The number of carriers at source physical node of physical path p (if BVTs are equipped at the physical node, then $c_k^i(p_s) = 1$) required by virtual k in request i if it is mapped to physical path p
$c_k^i(p_d)$ (integer)	The number of carriers at destination physical node of physical path p (if BVTs are equipped at the physical node, then $c_k^i(p_d) = 1$) required by virtual k in request i if it is mapped to physical path p
η_{pl} (binary)	$\eta_{pl} = 1$ if p transverses plink l
C_n (integer)	The number of carriers of each transponder at physical node n, $C_n = 1$ if the transponder is BVT, $C_n = C$ if the transponder is SBVT
δ_n (integer)	The cost of one transponder at physical node n. $\delta_n = \Gamma$ if SBVTs are equipped at the physical node n; $\delta_n = \Theta$ if BVTs are equipped at the physical node n

Table 2 Notation of variables

Symbol	Meaning
x_{ijn} (binary)	$x_{ijn} = 1$ if virtual node j in request i is mapped to physical node n
y_{ikp}(binary)	$y_{ikp} = 1$ if virtual link k in request i is mapped to physical path p, where p could be a dummy path with source physical node and destination physical node being a same physical node
τ_{iks}(binary)	$\tau_{iks} = 1$ if the starting frequency slot for virtual link k in request i is frequency slot s
g_{iks}(binary)	$g_{iks} = 1$ if virtual link k in request i uses frequency slot s
z_{iknq}(binary)	$z_{iknq} = 1$ if virtual link k in request i uses transponder q at physical node n
ϕ_{ikls}(binary)	$\phi_{ikls} = 1$ if virtual link k in request i uses frequency slot s on physical link l
γ_{iknqs} (binary)	$\gamma_{iknqs} = 1$ if virtual link k in request i uses frequency slot s on the extra link with transponder q at physical node n
θ_{iknq} (integer)	the number of carriers required by virtual link k in request i on transponder q at physical node n
m_{nq} (binary)	$m_{nq} = 1$ if transponder q at physical node n is used by some request

(10) $\sum_s \tau_{iks} \le 1$ for all i, k
(11) $\sum_s g_{iks} \le S \sum_s \tau_{iks}$ for all i, k
(12) $\sum_{nq} z_{iknq} = 2 \sum_s \tau_{iks}$ for all i, k
(13) $\sum_q z_{iknq} \le x_{iS(k^i)n} + x_{iD(k^i)n}$ for all i, k, n
(14) $m_{nq} \le z_{iknq}$ for all i, k, n, q
(15) $\phi_{ikls} \ge \sum_p y_{ikp}\eta_{pl} + g_{iks} - 1$ for all i, k, l, s
(16) $\gamma_{iknqs} \ge z_{iknq} + g_{iks} - 1$ for all i, k, n, q, s
(17) $\sum_{ik} \phi_{ikls} \le 1$ for all l, s;
(18) $\sum_{ik} \gamma_{iknqs} \le 1$ for all n, q, s
(19) $\theta_{iknq} \ge \sum_p c_k^i(p_s) y_{ikp} + C(z_{iknq} - 1)$ for all i, k, n, q
(20) $\theta_{iknq} \ge \sum_p c_k^i(p_d) y_{ikp} + C(z_{iknq} - 1)$ for all i, k, n, q
(21) $\sum_{ik} \theta_{iknq} \le C_n$ for all n, q

Constraints (1) and (2) ensure that each virtual node is mapped to one of its candidate physical nodes. Constraint (3) ensures that each physical node's VM capacity is not exceeded. Constraint (4) ensures that each virtual link is mapped to a physical path. Constraints (5) and (6) ensure that the two end physical nodes ($S(p)$ and $D(p)$) of physical path p assigned to virtual link k must match the two physical nodes assigned to the two end virtual nodes ($S(k^i)$ and $D(k^i)$) of k. Constraints (7) and (8) ensure the spectrum contiguity constraint, where s_0 is the first frequency slot index. Constraint (9) ensures that the bit rate required by each virtual link is satisfied. Note that if the source physical node and destination physical node of p are the same physical node, then $b_k^i(p) = 0$, which makes $\sum_s g_{iks} = 0$ since no frequency slot is allocated. Constraints (10) and (11) ensure that each virtual link has one starting frequency slot if its two end virtual nodes are not mapped to the same physical node ($\sum_s g_{iks} > 0$), where S is the number of frequency slots on each physical link. Constraint (12) ensures that each virtual link uses one transponder each at the source and destination physical nodes (total 2) if its two end virtual nodes are mapped to different physical nodes; otherwise, no transponder is required. Constraint (13) ensures that if virtual link k of request i uses physical node n as its source node or destination node, at most one transponder on node n is allocated. Constraint (14) ensures that $m_{nq} = 1$ if at least one virtual link uses transponder q at physical node n. Constraint (15) ensures $\phi_{ikls} = 1$ if virtual link k uses physical link l and frequency slot s. Constraint (16) ensures $\gamma_{iknqs} = 1$ if virtual link k uses extra link with transponder q at physical node n and frequency slot s. Constraints (17) and (18) ensure that each frequency slot on each (extra) physical link can be assigned to at most one virtual link. Please note that constraints (16) and (18) can be only applied to extra link with SBVTs to reduce the ILP time complexity. Constraints (19) and (20) ensure that the number of carriers on transponder q at physical node n allocated to virtual link k (i.e., θ_{iknq}) depends on the number of required carriers (i.e., $\sum_p c^i_k(p_s)y_{ikp}$ and $\sum_p c^i_k(p_d)y_{ikp}$). If the source virtual node and destination virtual node of k are not mapped to the physical node n or the transponder q is not assigned to virtual link k (i.e., $z_{iknq} = 0$), then θ_{iknq} can be set to 0. Constraint (21) ensures that each transponder's carrier capacity is not exceeded.

If there is a constraint that no two virtual nodes of the same request can be mapped to a single physical node, then we need another constraint (22) $\sum_j x_{ijn} \le 1$ for all i, n.

3.2 Lower bounds

3.2.1 LB1

This bound is based on the facts that (1) each lightpath allocated to each virtual link requires 0 or more carriers of a transponder at the source and destination physical nodes to which the end virtual nodes are mapped, and (2) there are S frequency slots on each physical link. To achieve this lower bound, the cost of each carrier of an SBVT is estimated as Γ/C; the cost of each carrier of a BVT is Θ. (Please note that there is only one carrier of a BVT.)

For each virtual link k of request i, according to the distance of each candidate physical path p^i_k in set P^i_k (in turn, the modulation format of each physical path) and the virtual link's bit rate requirement Λ^i_k, we can calculate the number of required frequency slots as $b^i_k(p)$ (including guardband), the number of required carriers $c^i_k(p_s)$ at source node of physical path p^i_k, the number of required carriers $c^i_k(p_d)$ at destination node of physical path p^i_k, and in turn the transponder cost $\delta^k(p)$ that need to be allocated to virtual link k of request i if it is mapped to physical path p (calculated based on the number of required carriers and type of transponders). ($b^i_k(p)$, $c^i_k(p_s)$, $c^i_k(p_d)$, and $\delta^k(p)$ are 0 if p is a dummy physical path with same end physical nodes. For virtual link k of request i, we can find the minimum transponder cost that has to be allocated as $\hat{\delta}_{ik} = \min_p \delta^k(p)$. In addition, $b^i_k(q)$ frequency slots are required on every physical link l along the physical path q_{ik} that achieves $\hat{\delta}_{ik}$. If this minimum cost allocation can be done with all virtual links, the cost will be $\sum_{ik} \hat{\delta}_{ik}$, but this might not be achievable due to spectrum limit as explained below.

For each physical link l, there are several physical paths using frequency slots on it (according to the above minimum cost allocation). Each physical path q_{ik} using l requires the corresponding minimum frequency slots $b^i_k(q)$. For each physical path q_{ik}, among all the candidate physical paths for k, find the minimum transponder cost g_{ikl} if physical link l is *not* used by k. Let $v^k_q = g_{ikl} - \hat{\delta}_{ik}$ (i.e., if virtual link k does not use physical link l, v^k_q more cost is required). However, since there is a spectrum limit on physical link l, it might not be possible to accommodate all the required $b^i_k(q)$ of all virtual links using physical link l to achieve their minimum cost assignment. As a result, some of these virtual links have to be allocated to some other physical links (resulting in at least v^k_q more cost for virtual link k); then, the task is to select these virtual links in order to minimize the resulting cost. Therefore, we now formulate a 0/1 knapsack problem (which can be solved easily) to obtain $LB1$. Let the capacity of the knapsack be $S + G$ (G is guardband); there are several items with weight $b^i_k(q)$ and value v_q. The objective of 0/1 knapsack is to maximize the total value of the knapsack with the weight constraint. Suppose the result is Z_l; this means that $y_l = \sum_q v_q - Z_l$ more cost is required to satisfy all requests. Then, $LB1 = \sum_{ik} \hat{\delta}_{ik} + \sum_l y_l$. The time complexity to compute $LB1$ is $O(IKV^2M + LX(S + G))$, where I is the number of requests, K is the maximum number of virtual links per request, V is the number of candidate physical nodes per virtual node, M is the number of physical paths for each pair of physical nodes, and L is the number of physical links in the network.

3.2.2 LB2

$LB2$ is obtained by observing that (1) virtual links that share an end virtual node must be allocated transponders at a single physical node (the physical node to which the end virtual node is mapped) and (2) capacity constraint of each physical node (H VMs). Consider a virtual node j which has several candidate physical nodes $n_1, n_2, n_3, \ldots$. There are several virtual links $k_1, k_2, k_3, \ldots$ that share j as one end virtual node. If j is mapped to n_1, then some carriers of transponders at physical node n_1 will be allocated to $k_1, k_2, k_3, \ldots$. We can find the minimum transponder cost for these virtual links if j is mapped to n_1 based on the candidate physical paths of $k_1, k_2, k_3, \ldots$. Denote $Q_j(n_1)$ as the corresponding transponder cost. Similarly, $Q_j(n_2), Q_j(n_3), \ldots$ can be calculated. Therefore, we can find the best physical node mapping for virtual node j in terms of the required cost $\bar{\delta}_j = \min(Q_j(n_1), Q_j(n_2), Q_j(n_3), \ldots)$. If this minimum cost allocation can be done for all virtual nodes, the cost will be $\sum_j \bar{\delta}_j$, but this might not be achievable due to VM limit as explained below.

The idea is similar to that in $LB1$. For each physical node n, there are several virtual nodes j using VMs on it (according to the above allocation). Each virtual node has a VM requirement h_j. For all virtual links with end virtual node j, find the minimum required cost g_{jn} for these virtual links if virtual node j is *not* mapped to physical node n. Let $v_j = g_{jn} - \bar{\delta}_j$ (i.e., if j is not mapped to n, v_j more cost is required). However, there is a VM limit for physical node n, and it might not be possible to accommodate all the required h_j of all virtual nodes mapping to physical node n to achieve their minimum cost assignment. As a result, some of these virtual nodes have to be allocated to some other physical nodes (resulting in at least v_j more cost for virtual node j); then, the task is to select these virtual nodes in order to minimize the resulting cost. We again have a 0/1 knapsack problem for physical node n. The capacity of the knapsack is H (the VM capacity), and there are several items, with item j having weight h_j and value v_j. The objective is to maximize the total value of the knapsack with the weight constraint. Suppose the result is Z_n, which means $y'_n = \sum_j v_j - Z_n$ more cost is required to satisfy all requests. Then, $LB2 = \sum_j \bar{\delta}_j + \sum_n y'_n$. The time complexity to compute $LB2$ is $O(IKV^2M + NIJH)$, where N is the number of physical nodes, J is the maximum number of virtual nodes per request, and H is the number of VMs per physical node.

3.3 Proposed heuristics

Although the ILP can provide an optimal solution for all VN requests, it only works for small instances due to high time complexity. For larger instances, we propose three heuristics based on list scheduling and then propose three improved meta-heuristics based on Tabu search. Note that since the VNM problem is NP-complete, the heuristics may occasionally fail to produce a result even though a valid mapping exists.

3.3.1 VN-LL heuristic

This heuristic first assigns a weight W_i to each request i based on its requirement; then, requests are sorted by decreasing order of weights and mapped one by one (Steps 1 and 4). Regarding a particular virtual request i under consideration, a weight ω_{ik} is assigned to each virtual link k^i, and the virtual links are sorted by decreasing order of weights and are mapped following this order (Steps 4.a and 4.b). For virtual link k^i being mapped, every possible candidate (as a combination of a physical path, a band of contiguous frequency slots, and two transponders at source and destination physical nodes, respectively) is assigned a weight, and the combination with minimum weight is allocated to k^i (Steps 4.b(1)–4.b(4)).

Step 1: Assign weight $W_i = \max\left(\frac{\hat{h}_i}{\bar{h}\mu}, \frac{\hat{\Lambda}_i}{\bar{\Lambda}\nu}\right)$ to each VN request i according to its VM and bit rate requirement; then sort the requests in decreasing order of weights
Step 2: TrivialRequestCheck() (used to map an entire VN request into a single physical node if possible)
Step 3: CheckOneCandidate() (used to map virtual nodes/virtual links with only one physical node/path candidate to map to)
Step 4: Schedule ordered unmapped requests one by one
foreach sorted request i **do**
  Step 4.a: Assign weight $\omega_{ik} = \bar{b}_k^i / m_i^k$ to each unmapped virtual link k of i according to its frequency slot requirement and number of candidate paths, then sort the unmapped virtual links in decreasing order of weights
  Step 4.b: Schedule ordered unmapped virtual links of i one by one
  foreach unmapped virtual link k of request i **do**
    foreach allocation-combination x (including candidate physical path p, frequency slot band f on p, transponder $t_s(p)$ at source physical node of p, transponder $t_d(p)$ at destination physical node of p) **do**
      Step 4.b(1): Calculate Δ_x the weight of x according to the required VMs, frequency slots, and carriers of allocation-combination x
    end
    Step 4.b(2): Map virtual link k to allocation combination $\bar{x}$ that achieves $\min_x \Delta_x$
    Step 4.b(3): Update VMs, frequency slots, carriers resources at the corresponding physical nodes, physical links and transponder with the virtual link mapping
    Step 4.b(4): CheckOneCandidate() (used to map virtual nodes/virtual links with only one physical node/path candidate to map to)
  end
end

Algorithm 1: VN-LL algorithm

In Step 1, weight W_i of request i is calculated as $W_i = \max\left(\frac{\hat{h}_i}{\bar{h}\mu}, \frac{\hat{\Lambda}_i}{\bar{\Lambda}\nu}\right)$, where $\hat{h}_i = \sum_j h_j^i$ is the total number of VMs required by i, $\hat{\Lambda}_i = \sum_k \Lambda_k^i$ is the total bit rate requirement of i, $\bar{h} = \sum_i \hat{h}_i$, $\bar{\Lambda} = \sum_i \hat{\Lambda}_i$, $\mu = \sum_{ij} f_j^i$ where f_j^i is the number of physical nodes the virtual node j can be mapped to, and $\nu = \sum_{ik} r_k^i$ where r_k^i is the number of candidate physical paths the virtual link k can be mapped to. Decreasing order of request weights gives higher priority to resource-intensive requests.

Before mapping the requests, we check (in Step 2 *TrivialRequestsCheck*) whether any request i can be mapped to a physical node n without affecting the virtual node mapping of other requests (i.e., all other virtual nodes can still be mapped to n if n is one of their candidate physical nodes). It is best to map an entire request to a single physical node if possible since this mapping will not use any frequency slots nor transponders.

There is another check made before mapping VN requests and after every virtual link mapping (Steps 3 and 4.b(4)). There is a VM limit at every physical node and a frequency slot limit on each physical link; the purpose of *CheckOneCandidate* is to share these limited resources fairly by first mapping virtual nodes and virtual links that have only one physical node and physical path to map to. There are two steps for *CheckOneCandidate* function. In the first step, all candidate physical nodes for each unmapped virtual node are found by taking into account the current number of available VMs at each physical node. If there is only one possible candidate physical node for a virtual node, assign the physical node to it, and update VMs at the physical node. In the second step, all candidate physical paths for each unmapped virtual link are found by taking into account the current number of available frequency slots and the possible candidate physical nodes (with VM availability consideration) of the virtual link's end virtual nodes. (Note that some end virtual nodes might have been mapped already in previous steps.) If there is only one possible candidate physical path for a virtual link, or all of its candidate physical paths bypass any common physical link, then the virtual link will be mapped first. After mapping these virtual links, the corresponding VMs, transponder resources, and frequency slots availability are updated.

For a particular request i under consideration, each of its unmapped virtual links k is assigned a weight ω_{ik} (Step 4.a) as $\bar{b}_k^i / m_i^k$, where m_i^k is the number of possible candidate physical paths for k, and $\bar{b}_k^i$ is the current minimum number of required frequency slots for k among the m_i^k candidate physical paths. The virtual links of i are sorted in decreasing order of weights to first map virtual links requiring more frequency slots and having fewer candidate physical paths.

Now, we describe the virtual link mapping steps (Steps 4.b(1)–4.b(4)). For each unmapped virtual link k (with source virtual node $s(k)$ and destination virtual node $d(k)$) mapping, all of its currently available candidate x, as a combination of physical path p (with source physical node $s(p)$ and destination physical node $d(p)$), a band of frequency slots f on p, transponder $t_s(p)$ at $s(p)$, and transponder $t_d(p)$ at $d(p)$, will be assigned a weight $\Delta_x = h'_x + t'_x + b_k(p)/S$, where $h'_x = \max(m_{s(k)}h_{s(k)}/H'_{s(p)}, m_{d(k)}h_{d(k)}/H'_{d(p)})$ if $s(p) \neq d(p)$, otherwise $h'_x = (m_{s(k)}h_{s(k)} + m_{d(k)}h_{d(k)})/H'_{s(p)}$, where $m_{s(k)} = 0$ ($m_{d(k)} = 0$) if $s(k)$ ($d(k)$) has already been mapped, otherwise they are set as 1; $h_{s(k)}$ ($h_{d(k)}$) is the number of required VMs for $s(k)$ ($d(k)$), and $H'_{s(p)}$ ($H'_{d(p)}$) is the number of currently available VMs at $s(p)$ ($d(p)$); t'_x is the cost of allocated transponder (according to the amount of allocated carriers of SBVTs and BVTs) to virtual link k; $b_k(p)$ is the numbers of required frequency slots if k is mapped to p; S is the frequency slots capacity of each plink. Virtual link k is mapped to the candidate achieving the minimum value $\Delta = \min_x \Delta_x$. After that, VM, transponders (with carriers), and frequency slot resources are updated for the corresponding physical links and physical nodes (allocated to virtual link k and its end virtual nodes). At last, *CheckOneCandidate* function is called since the mapping of virtual link k might result in changes of other virtual nodes'/virtual links' available mapping candidates. This procedure is repeated until all virtual links of virtual request i are mapped.

Note that if no two virtual nodes of the same request can be mapped to a single physical node, then there is no TrivialRequestCheck() function. In addition, dummy paths (with same end physical nodes) will not be considered as candidate physical paths in all steps. The time complexity of VN-LL is $O(I^3J^2NK + V^2MSI^3K^3)$.

3.3.2 ALL-LL

ALL-LL heuristic is similar to VN-LL heuristic with the difference that VN-LL first sorts VN requests, and then does the virtual link mapping for these requests one by one. In ALL-LL heuristic, all virtual links of all requests are sorted by decreasing order of their weights ω_{ik} (defined in VN-LL heuristic), and the virtual links are then mapped one by one as shown in Algorithm 2. The time complexity of ALL-LL is the same as VN-LL.

```
Steps 1-3 of VN-LL algorithm
Assign weight ω_k to each unmapped virtual link k according to its frequency slot
  requirement and number of candidate paths, then sort the unmapped virtual links
  in decreasing order of weights
Schedule ordered unmapped virtual links one by one
foreach unmapped virtual link k do
  | foreach allocation-combination x (including candidate physical path p,
  |   frequency slot band f on p, transponder t_s(p) at source physical node of p,
  |   transponder t_d(p) at destination physical node of p) do
  |   | Step 4.b(1)
  | end
  | Steps 4.b(2)-b(4)
end
```

Algorithm 2: ALL-LL algorithm

Fig. 3 Google datacenter network

3.3.3 LB-LL

The LB-LL heuristic is derived according to the idea of LB2 as shown in Algorithm 3.

```
Steps 1-3
Step 4: Schedule ordered unmapped requests one by one
foreach sorted request i do
    while There exist unmapped virtual nodes in request i do
        Assign weight τ_ij to each unmapped virtual node j of i
        Map the umapped virtual node with maximum weight to the physical node
          with minimum transponder cost
        Update the VM resources on the physical node
    end
    Steps 4.a and 4.b
end
```

Algorithm 3: LB-LL algorithm

Steps 1–3 are the same as VN-LL heuristic. For each virtual request under consideration, its virtual nodes are first mapped, and then the virtual links are mapped accordingly. The weight τ_{ij} assigned to each unmapped virtual node *j* of request *i* is set as follows: For each possible candidate physical node *n* (with sufficient VM resources) of virtual node *j*, find the minimum amount of required transponder cost δ_{jn} at physical node *n* for all the virtual links ending with virtual node *j*; τ_{ij} is set as the difference between the minimum δ_{jn} and the second minimum $\delta_{jn'}$. Then, the unmapped virtual node j' with maximum weight is selected and mapped to the physical node achieving $\min_n \delta_{j'n}$. For the virtual link mapping, the steps are the same as VN-LL heuristic. The time complexity of LB-LL is the same as VN-LL.

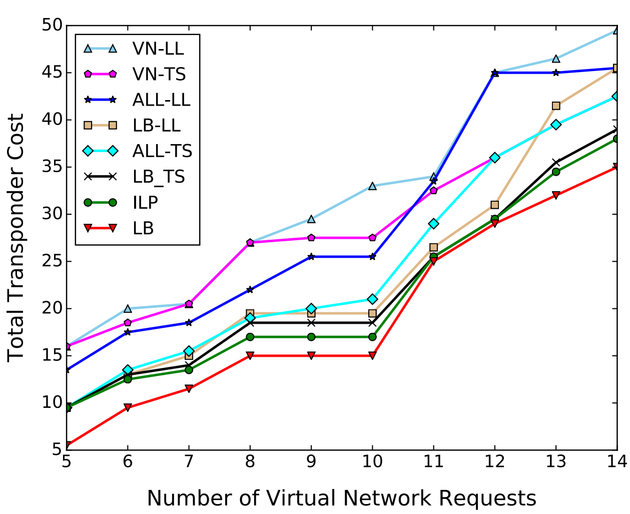

Fig. 4 Transponder cost in 6-node network scenario 1

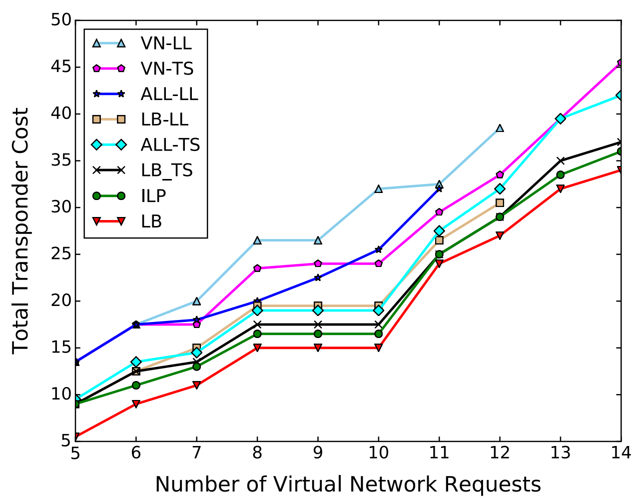

Fig. 5 Transponder cost in 6-node network scenario 2

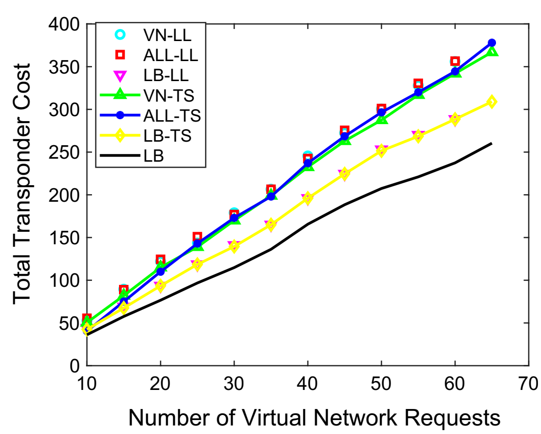

Fig. 6 Transponder cost versus number of VN requests for scenario 1

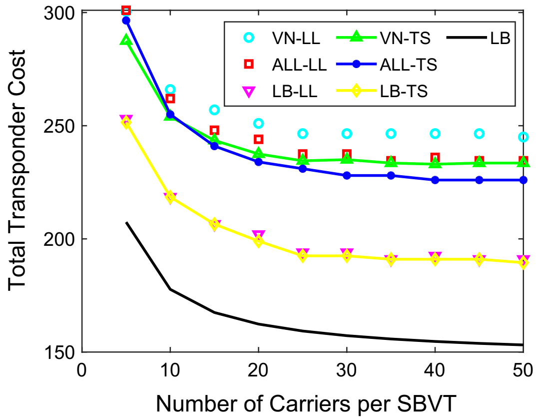

Fig. 7 Transponder cost versus number of carriers per SBVT for scenario 1

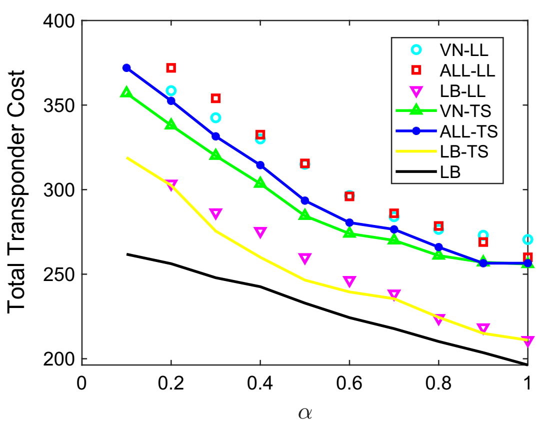

Fig. 8 Transponder cost versus α for scenario 1

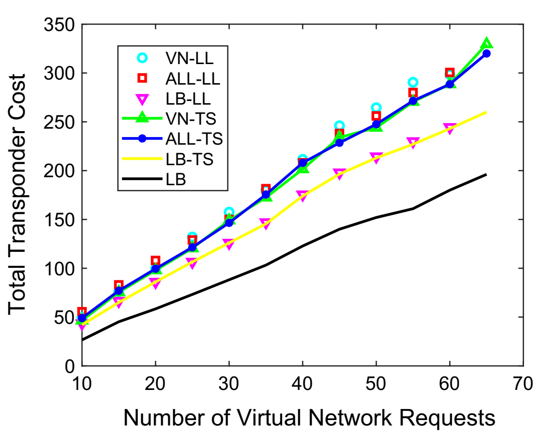

Fig. 9 Transponder cost versus number of VN requests for scenario 2

3.3.4 Tabu search

We utilize Tabu search meta-heuristic to improve the VN-LL, ALL-LL, and LB-LL heuristics performance,

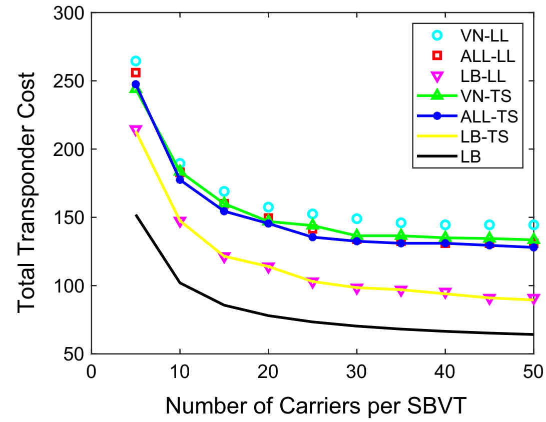

Fig. 10 Transponder cost versus number of carriers per SBVT for scenario 2

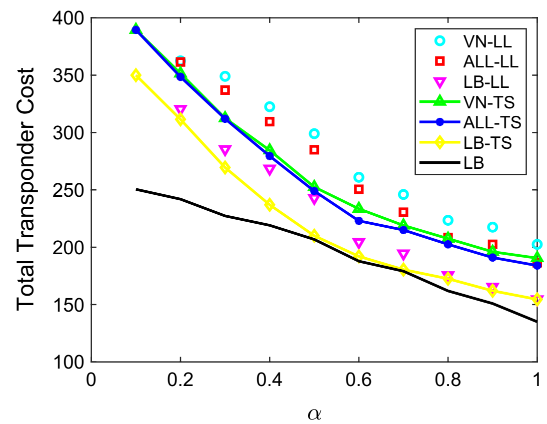

Fig. 11 Transponder cost versus α for scenario 2

called VN-TS, ALL-TS, and LB-TS, respectively. The main idea of Tabu search is to utilize a neighborhood search procedure to iteratively generate a better solution from one potential solution by exploring the neighborhood of current potential solution, until an attempt limit. In VN-TS/LB-TS algorithm (ALL-TS algorithm), the Tabu search method is utilized to iteratively generate different ordered sets of VN requests (virtual links), and each order is treated as a Tabu search solution. Given this order, the remaining steps in VN-LL/LB-LL (ALL-LL) are kept the same to find the transponder cost. The initial request (vlink) order is the one used in VN-LL/LB-LL (ALL-LL). To create a new ordered set based on current order, two VN requests (virtual links) in current order are randomly selected and swapped. To improve performance, ξ different new ordered sets are created for each current order (according to different swaps). The new order that results in minimum transponder cost will be used as current order for the next iteration. The procedure is repeated for Φ iterations. The larger the ξ and Φ values, the better the performance, but the longer simulation time. The time complexity is $O(\xi\Phi I^3 J^2 NK + \xi\Phi V^2 MSI^3 K^3)$.

4 Simulation results

We present results for two network topologies: the small 6-node network, as shown in Fig. 2, and Google datacenter network [19], as shown in Fig. 3. There are two scenarios with each network topology. In the first scenario, one third of the physical nodes are equipped with SBVTs, and in the second scenario, two thirds of the physical nodes are equipped with SBVTs.

There are three modulation levels used in the simulations: DP-QPSK, DP-8QAM, and DP-16QAM, and their corresponding transmission reach limits are 3000 km, 1000 km, and 650 km, respectively [15]. The spectrum efficiencies (in Gbps/GHz) of the three modulation formats are: 4, 6, and 8. The guardband is set as $G = 1$ frequency slot. The bandwidth of each carrier in SBVT is 25 GHz [16]. The capacity of SBVT and BVT is the same. The cost of each BVT is set as 1, while the cost of each SBVT is set as 1.5 [20–23]. Each virtual network request has either three or four virtual nodes (selected randomly). Each virtual node has randomly 1–3 physical node candidates. The probability that there is a virtual link between each pair of virtual nodes is set as 0.8. This high value is selected since only connected graphs are used as virtual network requests for simulation, which makes it very likely to generate connected graphs. The bit rate requirement for each virtual link is uniformly randomly set in the range 100–500 Gbps.

4.1 Numerical results for 6-node network

In this section, we compare the results of ILP, lower bounds, and six heuristics for the small 6-node network topology. We assume that each physical node has 30 VMs and there are 30 frequency slots on each physical link. Each virtual node requires a random number between 1 and 5 VMs. The number of carriers per SBVT is set as $C = 5$. Let I denote the number of virtual network requests. The value of ξ is set as $I(I-1)/2$, and the value of Φ is set as 500 for Tabu search algorithms. Sample results for 1–14 virtual network requests are shown in Figs. 4 and 5, where LB is the minimum of $LB1$ and $LB2$.

It can be seen that LB-LL algorithm outperforms ALL-LL and VN-LL algorithms. Tabu search can improve the heuristic results, suggesting that the order of requests (or virtual links) can affect the performance especially for the cases that the resources requirements are close to the resources capacity. The LB values are not far away from the optimal ILP results. By comparing the two scenarios, it can be seen that the more the physical nodes with SBVTs, the lower the transponder cost.

4.2 Numerical results for Google network

For the larger Google network topology, we show the results for lower bounds and six heuristics as the average of ten seeds simulation results. We assume that each physical node has 1000 VMs, and there are 320 frequency slots on each link (as optical C-band is 4000 GHz and each frequency slot is 12.5 GHz). The number of required VMs for each virtual node is randomly selected in $\{10, 20, 30, 40, 50\}$. The value of ξ is set as 20, and the value of Φ is set as 25. Figures 6, 7, and 8 are the results for scenario 1, and Figs. 9, 10, and 11 are the results for scenario 2. Figures. 6 and 9 show the transponder cost as a function of the number of requests when there are five carriers per SBVT. Figures 7 and 10 show the transponder cost for 50 requests versus the number of carriers per SBVT. Figures 8 and 11 show simulation results for mixed virtual links' bit rate requirements of 50 requests, where α percentage of virtual links have bit rate requirement in range 100–200 Gbps, $1-\alpha$ percentage of virtual links have bit rate requirement in range 400−500 Gbps (when there are five carriers per SBVT). These results confirmed that performance of LB-LL is better than the performance of VN-LL and ALL-LL. Tabu search produces feasible solutions when VN-LL, ALL-LL, and LB-LL fail. The differences between Tabu search and lower bound are on average 11.6%, 18.5%, 8.8%, 16.7%, 18.8%, and 8.8% in these figures, respectively.

5 Conclusions

A virtual network mapping problem is investigated in elastic optical inter-datacenter networks equipped with sliceable transponders. The objective is to minimize the total transponder cost. We proposed an Integer Linear Programming formulation, two lower bounds, three heuristics based on list scheduling, and three meta-heuristics based on Tabu search. Simulation results show that the Tabu search variants of the proposed heuristics achieve quite good performance compared to the derived lower bounds.

Acknowledgements This work was supported in part by NSF Grants CNS-1813617 and CNS-1813772.

References

1. Tomkos, I., Azodolmolky, S., Sole-Pareta, J., Careglio, D., Palkopoulou, E.: A tutorial on the flexible optical networking

paradigm: state of the art, trends, and research challenges. Proc. IEEE **102**(9), 1317–1337 (2014)
2. Jinno, M., Kozicki, B., Takara, H., Watanabe, A., Sone, Y., Tanaka, T., Hirano, A.: Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network. IEEE Commun. Mag. **48**(8), 138–145 (2010)
3. Chowdhury, N.M.M.K., Rahman, M.R., Boutaba, R.: Virtual network embedding with coordinated node and link mapping. In: 28th Conference on Computer Communications (INFOCOM), pp. 783–791. IEEE, Rio de Janeiro, Brazil (2009)
4. Zhao, J., Subramaniam, S., Brandt-Pearce, M.: Virtual topology mapping in elastic optical networks. In: International Conference on Communications (ICC), pp. 3904–3908. IEEE, Budapest, Hungary (2013)
5. Zhang, S., Shi, L., Vadrevu, C.S.K., Mukherjee, B.: Network virtualization over WDM and flexible-grid optical networks. Opt. Switch. Netw. **10**(4), 291–300 (2013)
6. Gong, L., Zhu, Z.: Virtual optical network embedding (VONE) over elastic optical networks. J. Lightw. Technol. **32**(3), 450–460 (2014)
7. Wang, Y., McNulty, Z., Nguyen, H.: Network virtualization in spectrum sliced elastic optical path networks. J. Lightw. Technol. **35**(10), 1962–1970 (2017)
8. Madani, F.M., Mokhtari, S.: Fragmentation-aware load-balancing virtual optical network embedding (VONE) over elastic optical networks. In: The Sixth International Conference on Cloud Computing. GRIDs, and Virtualization, pp. 27–32. Nice, France (2015)
9. Yu, C., Guo, L., Hou, W.: Novel elastic optical network embedding using re-optimized VCAT framework accompanied by hitless PPSM function. J. Lightw. Technol. **34**(22), 5199–5213 (2016)
10. Soto, P., Botero, J.F., Hesselbach, X.: Optimal occupancy mapping of virtual networks over elastic optical infrastructures. In: 19th International Conference on Transparent Optical Networks (ICTON), pp. 1–7. Girona, Spain (2017)
11. Xie, W., Jue, J., Zhang, Q., Wang, X., She, Q., Palacharla, P., Sekiya, M.: Survivable virtual optical network mapping in flexible-grid optical networks. In: International Conference on Computing. Networking and Communications (ICNC), pp. 221–225. Honolulu, HI, USA (2014)
12. Chen, B., Zhang, J., Xie, W., Jue, J., Zhao, Y., Shen, G.: Cost-effective survivable virtual optical network mapping in flexible bandwidth optical networks. J. Lightw. Technol. **34**(10), 2398–2412 (2016)
13. Zhu, M., Gao, P., Zhang, J., Zeng, X., Zhang, S.: Energy efficient dynamic virtual optical network embedding in sliceable-transponder-equipped EONs. In: IEEE Global Communications Conference, pp. 1–6. Singapore (2017)
14. Dallaglio, M., Giorgetti, A., Sambo, N., Velasco, L., Castoldi, P.: Routing, spectrum, and transponder assignment in elastic optical networks. J. Lightw. Technol. **33**(22), 4648–4658 (2015)
15. Martinez, R., Casellas, R., Vilalta, R., Munoz, R.: GMPLS/PCE-controlled multi-flow optical transponders in elastic optical networks. J. Opt. Commun. Netw. **7**(11), B71–B80 (2015)
16. Zhang, J., Ji, Y., Song, M., Zhao, Y., Yu, X., Zhang, J., Mukherjee, B.: Dynamic traffic grooming in sliceable bandwidth-variable transponder-enabled elastic optical networks. J. Lightw. Technol. **33**(1), 183–191 (2015)
17. Zhao J., Subramaniam, S.: Embedding virtual networks in flexible optical networks with sliceable transponders. In: Conference On Optical Network Design And Modeling. Athens, Greece (2019)
18. Christodoulopoulos, K., Tomkos, I., Varvarigos, E.A.: Elastic bandwidth allocation in flexible OFDM-based optical networks. J. Lightw. Technol. **29**(9), 1354–1366 (2011)
19. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Holzle, U., Stuart, S., Vahdat, A.: B4: experience with a globally-deployed software defined wan. In: ACM SIGCOMM conference, pp. 3–14. Hong Kong, China (2013)
20. Pan, Y.: Joint resource provisioning in optical cloud networks. In: Ph.D. Dissertation. University of Nebraska, Lincoln (2016)
21. Tanaka, T., Hirano, A., Jinno, M.: Performance evaluation of elastic optical networks with multi-flow optical transponders. In: European conference and exhibition on optical communications (ECOC). Amsterdam, Netherlands (2012)
22. Tanaka, T., Hirano, A., Jinno, M.: Advantages of IP over elastic optical networks using multi-flow transponders from cost and equipment count aspects. Opt. Exp. **22**(1), 62–70 (2014)
23. Christodoulopoulos, K., Varvarigos, E.: Evaluating flexibility degrees in optical networks. In: Optical Fiber Communication Conference. Los Angeles, USA (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dr. Juzi Zhao is an Assistant Professor of Department of Electrical Engineering at San Jose State University. Her main areas of research are optical networking and network security. She received her PhD in Electrical Engineering from The George Washington University in 2014. She worked as a Postdoctoral Associate at University of Massachusetts Lowell and Chalmers University of Technology.

Suresh Subramaniam (S'95-M'97-SM'07-F'15) received the PhD degree in electrical engineering from the University of Washington, Seattle, in 1997. He is a Professor in and Chair of the Department of Electrical and Computer Engineering at the George Washington University, Washington, DC. His research interests are in the architectural, algorithmic, and performance aspects of communication networks, with current emphasis on optical networks, cloud computing, datacenter networks, and IoT. He has published over 200 peer-reviewed papers. Dr. Subramaniam is a co-editor of three books on optical networking. During 2012 and 2013, he served as the Chair of the IEEE ComSoc Optical Networking Technical Committee. He received the 2017 SEAS Distinguished Researcher Award at GWU, and he is a Fellow of the IEEE.