



Column generation algorithms for virtual network embedding in flexi-grid optical networks

RONGPING LIN,^{1,*} SHAN LUO,² JINGWEI ZHOU,¹ SHENG WANG,¹
BIN CHEN,³ XIAONING ZHANG,¹ ANLIANG CAI,⁴ WEN-DE ZHONG,⁵
AND MOSHE ZUKERMAN⁴

¹*School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, 611731, China*

²*School of Astronautics and Aeronautic, UESTC, Chengdu, Sichuan, 611731, China*

³*College of Information Engineering, Shenzhen University, Shenzhen, China*

⁴*Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China*

⁵*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

*linrp@uestc.edu.cn

Abstract: Network virtualization provides means for efficient management of network resources by embedding multiple virtual networks (VNs) to share efficiently the same substrate network. Such virtual network embedding (VNE) gives rise to a challenging problem of how to optimize resource allocation to VNs and to guarantee their performance requirements. In this paper, we provide VNE algorithms for efficient management of flexi-grid optical networks. We provide an exact algorithm aiming to minimize the total embedding cost in terms of spectrum cost and computation cost for a single VN request. Then, to achieve scalability, we also develop a heuristic algorithm for the same problem. We apply these two algorithms for a dynamic traffic scenario where many VN requests arrive one-by-one. We first demonstrate by simulations for the case of a six-node network that the heuristic algorithm obtains very close blocking probabilities to exact algorithm (about 0.2% higher). Then, for a network of realistic size (namely, USnet) we demonstrate that the blocking probability of our new heuristic algorithm is about one magnitude lower than a simpler heuristic algorithm, which was a component of an earlier published algorithm.

© 2018 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

OCIS codes: (060.4256) Networks, network optimization; (060.4254) Networks, combinatorial network design.

References and links

1. Cisco Inc., "Cisco visual networking index: forecast and methodology, 2016-2021" (2016).
2. N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.* **54**(5), 862–876 (2010).
3. N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. INFOCOM*, 783–791 (2009).
4. X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Comput. Commun. Rev.* **14**(2), 39–47 (2011).
5. H. Cui, S. Tang, X. Huang, J. Chen, and Y. Liu, "A novel method of virtual network embedding based on topology convergence-degree," in *Proc. Int. Conf. Commun.*, 246–250 (2013).
6. L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.* **24**(6), 3648–3661 (2016).
7. A. Jarray and A. Karmouch, "Decomposition approaches for virtual network embedding with one-shot node and link mapping," *IEEE/ACM Trans. Netw.* **23**(3), 1012–1025 (2015).
8. F. Zhou, M. Ju, and A. Ait-Ouahmed, "Joint optimization for multicast provisioning in mixed-line-rate optical networks with column generation approach," *J. Lightwave Technol.*, **36**(3), 637–649 (2018).
9. I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's," *IEEE Trans. Commun.* **40**(7), 1171–1182 (1992).
10. M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies," *IEEE Commun. Mag.* **47**(11), 66–73 (2009).

11. A. Cai, J. Guo, R. Lin, G. Shen, and M. Zukerman, "Multicast routing and distance-adaptive spectrum allocation in elastic optical networks with shared protection," *J. Lightwave Technol.* **34**(17), 4076–4088 (2016).
12. M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," *IEEE Commun. Mag.* **48**(8), 138–145 (2010).
13. A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: a survey," *IEEE Commun. Surv. Tutorials* **15**(4), 1888–1906 (2013).
14. A. Pages, J. Perello, S. Spadaro, J. A. Garcia-Espin, J. Ferrer Riera, and S. Figuerola, "Optimal allocation of virtual optical networks for the future Internet," in *Proc. Int. Conf. Opt. Netw. Des. Model.*, 1–6 (2012).
15. L. Gong, W. Zhao, Y. Wen, and Z. Zhu, "Dynamic transparent virtual network embedding over elastic optical infrastructures," in *Proc. Int. Conf. Commun.*, 3466–3470 (2013).
16. L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightwave Technol.* **32**(3), 450–460 (2014).
17. J. Zhao and M. Pearce, "Virtual topology mapping in elastic optical networks," in *Proc. Int. Conf. Commun.*, 3904–3908 (2013).
18. S. Zhang, L. Shi, C. Vadrevu, and B. Mukherjee, "Network virtualization over WDM and flexible-grid optical networks," *Optical Switching and Networking* **10**(4), 291–300 (2013).
19. M. R. Raza, M. Fiorani, A. Rostami, P. Ohlen, L. Wosinska, and P. Monti, "Dynamic slicing approach for multi-tenant 5G transport networks [invited]," *J. Opt. Commun. Netw.* **10**(1), A77–A90 (2018).
20. R. Lin, S. Luo, H. Wang, and S. Wang, "Energy-aware virtual network embedding in flexi-grid networks," *Opt. Express* **25**(24), 29699–29713 (2017).
21. L. R. Ford Jr. and D. R. Fulkerson, "A suggested computation for maximal multi-commodity network flows," *Management Science* **5**(1), 97–101 (1958).
22. G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research* **8**, 101–111 (1960).
23. G. L. Nemhauser, "Column generation for linear and integer programming," *Documenta Mathematica-Extra Volume ISMP* **5**(1), 65–73 (2012).
24. R. Lin, S. Luo, J. Zhou, S. Wang, A. Cai, W. D. Zhong, and M. Zukerman, "Virtual network embedding with adaptive modulation in flexi-grid networks," *J. Lightwave Technol.*, 1–13 (2017) (online access).
25. ILOG CPLEX, ILOG, Inc., Mountain View, CA[Online], Available: <http://www.ilog.com/products/cplex/>.

1. Introduction

The Cisco's 2016 white paper predicted that the global IP traffic will reach 9.3 Exabytes/day in 2021 [1]. This tremendous traffic will be generated by a variety of applications that impose a range of quality-of-service (QoS) requirements on enterprise, telecommunication and data center networks. In current network architectures, there is a certain inflexibility mismatch between these networks and the requirements of these applications which can be addressed by network virtualization technology [2] that hides the complexity by abstracting away substrate network details. This enables multiple virtual networks (VNs) each provides connectivity for a specific application to share the same substrate network and be isolated from each other. A VN consists of virtual nodes and virtual links, and the problem of virtual network embedding (VNE) is to allocate appropriate resources to VNs. Accordingly, VNE consists of *virtual node mapping* where a virtual node is mapped to a physical node and computational resource is reserved for the virtual node, and *virtual link mapping* where a virtual link is mapped to a physical path and bandwidth or spectrum is reserved along the path. A VNE example is provided in Fig. 1, where physical resources are shared by multiple VNs that provide heterogeneous services. In the figure, virtual links e_1 and e_2 of two different VNs are embedded into physical paths p_1 and p_2 , respectively, where a physical link is shared, and virtual nodes i and j are embedded in the physical node t .

The problem of VNE has been intensively investigated in electronic networks, e.g. Ethernet. Various strategies were provided to coordinate virtual node and virtual link mappings. In [3], an augmented substrate graph was proposed and the commodity flow algorithm was used to solve virtual link mapping. In [4], Markov random walk model was applied to rank network nodes. The ranking reflects the relative importance of nodes that can coordinate virtual node and virtual link mappings. In [5], virtual nodes were mapped to physical nodes within a geographical subgraph, which can reduce the bandwidth usage in virtual link mapping. In [6], algorithms based on a compatibility graph were proposed to solve the location-constrained VNE problem where constraints were added to limit the virtual node embedding to a restricted set of physical nodes.

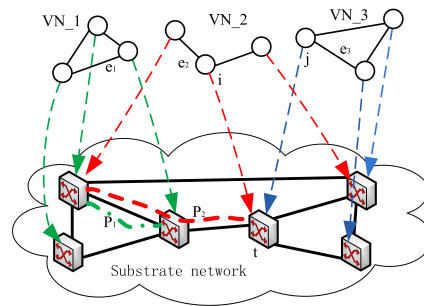


Fig. 1. A VNE example.

This problem had also been reduced to the minimum cost maximum clique problem in [6]. The work of [7] used the column generation method to solve the VNE problem considering a batch of VN requests that arrives simultaneously, and proposed an auction mechanism to control admission of VN requests. However, the solution is based on a set of given physical paths, which narrows the search space, but may lead to non-optimal solutions. All these algorithms cannot be directly applied to flexi-grid networks where the constraints are more complex than the bandwidth capacity of electronic networks. In [8], the column generate method was applied in the multicast provisioning problem in optical networks, and a multicast request is divided into multiple lightpath connections for simplicity. In this paper, the column generate method will be applied to a more complex problem, where a VN request has multiple connections and some connections may have the same source or destination node.

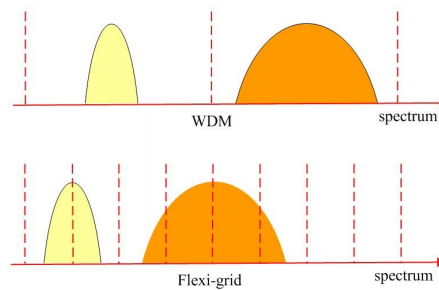


Fig. 2. Grid difference between WDM and flexi-grid.

Flexi-grid optical networks with a finer spectrum granularity than a wavelength division multiplexing (WDM) network can obtain high resource efficiencies. The WDM network allocates a fixed and coarse spectrum (a wavelength) to a lightpath that is an all-optical channel [9]. Meanwhile, in the spectrum domain, the centers of wavelengths are fixed and each wavelength has a fixed spectrum width (e.g. 50GHz). However, the flexi-grid network can allocate various number of spectrum slots, each of e.g. 6.25GHz, to a lightpath to achieve the flexibility of spectrum allocation [10]. For example, Figure 2 illustrates the spectrum grid differences between WDM network and flexi-grid optical network, where flexi-grid network can allocate just-enough spectrum resources for each lightpath while WDM network allocates a fixed amount of spectrum resources regardless of the lightpath bandwidth requirement. Similar to WDM networks, a lightpath in flexi-grid networks should satisfy the *continuity* constraint which ensures that all physical links of the lightpath should be assigned with the same spectrum band (from the same starting spectrum slot to same end slot). Note that the term *spectrum slot* used in this paper

is equivalent to the term *frequency slot* used in [11, 12]. In addition, the *contiguity* constraint ensures that the spectrum slots allocated to a link must be adjacent. These constraints add certain complexity to the VNE design in flexi-grid optical networks relative to electronic networks.

The VNE in electronic networks is known to be NP-hard [13]. Accordingly, VNE in flexi-grid networks must be NP-hard due to a higher complexity. The work in [14] considered virtual link mapping in flexi-grid optical networks without addressing virtual node mapping. In [15], a layered auxiliary graph was constructed with available spectrum resources, and used in optimization of VNE design. Then, a heuristic algorithm considering the degrees of physical nodes in node mapping step was also proposed. The authors of [15] extended their work in [16], where they proposed integer linear programming (ILP) formulations and heuristic algorithms for transparent and opaque VNE considering modulation selections. In transparent VNE, all physical paths for a VN are assigned with the same spectrum band, while in opaque VNE, lightpaths may have different spectrum bands. However, the virtual link embedding in opaque VNE was limited in a given physical paths, which may lead to a suboptimal solution. Another ILP formulation for VNE was proposed in [17], where a set of physical paths is given and the virtual link mapping is limited in these paths, and this may also lead to a suboptimal solution. An ILP formulation for VNE in flexi-grid optical networks with static traffic was provided in [18], where wavelength conversion was assumed at all network nodes and only three (40G, 100G and 400G) line rates (wavelength bit rates) are considered to simplify the problem. Accordingly, the ILP formulation and the algorithms of [18], cannot be applied to the problem addressed in this paper, where we consider a dynamic traffic scenario without wavelength conversion and a lightpath can provide the required bandwidth with just-enough spectrum resource. The authors of [19] considered both VNE problem and VN reconfiguration on 5G transport networks which are assumed to be WDM optical networks, and two ILP formulations were provided for these two problems. As in [18], wavelength conversion was assumed and the fixed line rate is considered for a wavelength due to the assumption of traditional WDM network. A heuristic algorithm was proposed based on enumerating all possible virtual node embedding choices, and selecting one with the least resource usages, which may not scale to cases where the network size becomes large. Accordingly, the ILP formulation and the algorithm are not applicable to the problem of this paper. Energy cost of VNE in flexi-grid networks has been considered in our recent paper [20], where an ILP and heuristic algorithms were proposed to reduce the electricity cost. However, the work in [20] is quite different to the work in this paper. In [20], the objective was to minimize energy consumption. To this end, power consumption of network and computation devices are considered, so the aim is to reduce the number of turn-on devices and increase efficiency in device sharing. By comparison, in this paper, we consider the VNE problem aiming to minimize embedding cost. Our heuristic algorithm, in addition to minimize cost, also seeks solutions that reduce the blocking probability of VN requests. In addition to the difference in objectives, the methodology used in this paper is different to that in [20]. In this paper, we use a column generation method that is known to speed up linear programming (LP) solutions [21–23] to solve the VNE problem, while in [20] the ILP was used. To the best of our knowledge, the problem in our paper has not been investigated by others, and we provide an efficient algorithm for this problem. We also investigated the VNE with adaptive modulation in flexi-grid networks [24], where transmission modulation was optimally selected to minimize spectrum usage. Our work in this paper considers the design cost of the VNE in flexi-grid optical networks, for which exact and heuristic algorithms using column generation are provided.

2. Virtual node and link embeddings

In general, the VNE problem consists of virtual node and virtual link embeddings. In flexi-grid networks, continuity and contiguity constraints are taken into consideration in the virtual link embedding.

$$F(i) = t, \quad i \in N^v, t \in N^s \quad (1)$$

In virtual node embedding, a physical node is selected to host a virtual node. Equation (1) indicates that physical node t hosts virtual node i , where N^v and N^s denote the set of virtual nodes and physical nodes, respectively. In addition the available computational resource of t is sufficient for the required computational resource of i .

$$G(e) = P_{F(d(e))}^{F(s(e))}, \quad e \in L^v \quad (2)$$

In virtual link embedding, a lightpath is set up to serve a virtual link. Equation (2) indicates that physical path $P_{F(d(e))}^{F(s(e))}$ is used to serve virtual link e , where $s(e)$, $d(e)$, $F(s(e))$ and $F(d(e))$ denote the source node of e , the destination node of e , the physical node hosting $s(e)$, and the physical node hosting $d(e)$, respectively, and L^v denotes the set of virtual links. Suppose $P_{F(d(e))}^{F(s(e))} = \{L_1, L_2, \dots, L_n\}$, where L_1, L_2, \dots, L_n are physical links and are cascaded to be the path. With the contiguity constraint, the spectrum slots allocated at any link of the path are contiguous, which forms a spectrum band. With the continuity constraint, the spectrum bands allocated along the path are the same, which forms an optical channel. A physical node that hosts a virtual node is the source or destination of lightpaths that provide connectivity for virtual links. Accordingly, virtual node embedding and virtual link embedding are dependent on each other.

3. Column generation formulation

This paper considers dynamic traffic scenarios, where VN requests arrive the network one-by-one over time. After their service is complete, the VN requests depart and are cleared from the system. For each arriving VN request, the formulation in this section can be applied to allocate network resources based on the unused resources at the time of the request arrival. If the existing network cannot accommodate the request due to the lack of resources, the request is blocked.

Based on the observation that most of the variables of an optimization problem will have a zero value in the final solution, the column generation method has been used to improve the time-efficiency of the problem solving [21–23]. In particular, the column generation method transforms an optimization problem into a restricted master problem (RMP) and a pricing problem (PP), and iteratively adds new variables that have negative reduced costs (RCs) into RMP. The RMP is a *small version* of the original problem, where the number of variables is reduced. PP checks whether the objective value of RMP can be further improved (exist negative reduced costs) by adding new variables (and correspondingly coefficient matrix columns of the variables). If a variable not in the RMP can improve the objective value, PP will add it to the RMP to form a new RMP. The iterations will not stop until the optimal result of the original problem is reached.

In this paper, PP is designed to consist of multiple PP-subproblems. Specifically, the embedding of a virtual link plus its two virtual end nodes is defined as a PP-subproblem, i.e., the PP-subproblem ij deals with the embedding of virtual link (i, j) plus two virtual nodes, tail i and head j . For a VN request, the number of PP-subproblems is equal to the number of virtual links. A solution of a PP-subproblem provides a *partial embedding solution* (PESs) to the entire VNE problem, then PESs are selected to form the solution of the RMP.

3.1. Restricted master problem

In the RMP, PESs are given and selected to form the solution.

Given:

$G^s(N^s, L^s)$: a directed graph denotes a substrate network. N^s and L^s denote the set of physical nodes and the set of physical links, respectively.

$G^v(N^v, L^v)$: a directed graph denotes a VN request. N^v and L^v denote the set of virtual nodes and the set of virtual links, respectively.

c_k : CPU unit cost at physical node k , $k \in N^s$.

c_{mn} : cost of a spectrum slot in physical link (m, n) , $(m, n) \in L^s$.

F : the spectrum slot index set of a physical link.

$U_{mn,t}$: the status of t th spectrum slot of physical link (m, n) , $(m, n) \in L^s$, $t \in F$. It equals to 1 when the slot is available, otherwise 0.

p_i : the CPU resources required by virtual node i , $i \in N^v$.

p_{ij} : the number of spectrum slots required by virtual link (i, j) , $(i, j) \in L^v$.

$Deg(i)$: degree of node i , which is equal to the summation of indegree and outdegree of node i .

$oadj(i)$: the set of head points adjacent to node i , so each node of $oadj(i)$ has a directed link to node i .

$iadj(i)$: the set of tail points adjacent to node i , so each node of $iadj(i)$ has a directed link to node i .

r : the index of an element of the set R^{ij} .

R^{ij} : the set of PESs provided by PP-subproblem ij . The r th element provides the virtual link, tail and head mapping information. In particular, it consists of $M_{mn,a}^{ij,r}$, $A_k^{ij,r}$, and $B_k^{ij,r}$. The size of R^{ij} is enlarged if new PESs are added into the RMP problem during iterations.

$M_{mn,a}^{ij,r}$: It takes the value of 1 when the r th PES provided by the PP-subproblem ij traverses physical link (m, n) and uses spectrum from a th slot, otherwise 0.

$A_k^{ij,r}$: It takes the value of 1 when the r th PES provided by the PP-subproblem ij has the tail i being mapped to the physical node k , otherwise 0.

$B_k^{ij,r}$: It takes the value of 1 when the r th PES provided by the PP-subproblem ij has the head j being mapped to physical node k , otherwise 0.

Variables:

y_r^{ij} : a Boolean variable which equals to 1 when the r th PES provided by PP-subproblem ij is selected in the RMP, otherwise 0. All the selected PESs are formed to be the optimal solution of the RMP. This variable will be linearized when the branch-and-bound approach is applied to solve the formulation.

Objective:

The objective of RMP is as follows:

$$\text{Min} \sum_{ij \in L^v} \sum_{r \in R^{ij}} \left[\sum_{mn \in L^s} \sum_{a \in F} c_{mn} p_{ij} M_{mn,a}^{ij,r} + \sum_{k \in N^s} c_k \left(p_i \frac{A_k^{ij,r}}{Deg(i)} + p_j \frac{B_k^{ij,r}}{Deg(j)} \right) \right] y_r^{ij} \quad (3)$$

which is to minimize the embedding cost that consists of spectrum cost and CPU cost. Specifically, for PP-subproblem ij , the spectrum cost and CPU cost of the selected PES r (y_r^{ij} is equal to 1) are calculated, then the total cost of all PP-subproblems provides the entire VNE cost.

The second term of (3) represents the CPU cost. It is noted that the embedding of a specific virtual node i is jointly decided by the $Deg(i)$ PP-subproblems that are with virtual node i as tail or head of virtual links, then node i should be embedded in the same physical node.

Constraints:

$$\sum_{r \in R^{ij}} y_r^{ij} = 1, \quad \forall ij \in L^v \quad (4)$$

Equation (4) ensures that for each PP-subproblem ij , only one PES in the set R^{ij} is selected.

There are $Deg(i)$ PP-subproblems that provide PESs containing the embedding of the virtual node i , and these PP-subproblems embed virtual node i into the same physical node that is the source or destination of $Deg(i)$ embedding lightpaths. The node degree consistency between the

virtual node and the embedding solution are ensured as follows.

$$\sum_{t \in \text{oad}(i)} \sum_{r \in R^{it}} A_k^{it,r} y_r^{it} + \sum_{t \in \text{iad}(i)} \sum_{r \in R^{ti}} B_k^{ti,r} y_r^{ti} = \text{Deg}(i) \sum_{r \in R^{ij}} A_k^{ij,r} y_r^{ij}, \quad \forall ij \in L^v, k \in N^s \quad (5)$$

$$\sum_{t \in \text{oad}(j)} \sum_{r \in R^{jt}} A_k^{jt,r} y_r^{jt} + \sum_{t \in \text{iad}(j)} \sum_{r \in R^{tj}} B_k^{tj,r} y_r^{tj} = \text{Deg}(j) \sum_{r \in R^{ij}} B_k^{ij,r} y_r^{ij}, \quad \forall ij \in L^v, k \in N^s \quad (6)$$

In (5), if a solution of PP-subproblem ij is selected to make tail i embedded in physical node k (i.e. $\sum_{r \in R^{ij}} A_k^{ij,r} y_r^{ij} = 1$), all PP-subproblems with node i as the tail of virtual link and all PP-subproblems with node i as the head of virtual link should embed node i in physical node k . Otherwise, the node degree consistency of node i is not held. Similarly, (6) ensures the node degree consistency of head j of PP-subproblem ij .

$$\sum_{ij \in L^v} \sum_{r \in R^{ij}} \left(\frac{A_k^{ij,r}}{\text{Deg}(i)} + \frac{B_k^{ij,r}}{\text{Deg}(j)} \right) y_r^{ij} \leq 1, \quad \forall k \in N^s \quad (7)$$

$$\sum_{ij \in L^v} \sum_{r \in R^{ij}} \sum_{\substack{t=a-p_{ij}+1, \\ a-p_{ij}+2, \dots, a}} M_{mn,t}^{ij,r} y_r^{ij} \leq U_{mn,a}, \quad \forall mn \in L^s, a \in F \quad (8)$$

Equation (7) ensures that a physical node can host one virtual node at most. Equation (8) ensures that any spectrum slot of a physical link can be occupied by at most one virtual link embedding, i.e. at most one selected PES can occupy the a th slot of physical link (m, n) . It is noted that for the spectrum requirement p_{ij} , all spectrum allocations with starting slot from $a - p_{ij} + 1$ th slot to a th slot contain the a th spectrum slot.

3.2. Pricing problem

As discussed above, PP consists of multiple PP-subproblems each of which provides PESs, and a PES is an embedding solution for a virtual link and its two end nodes. In addition, the objective function of PP-subproblem is the reduced cost of the RMP variables. The following ILP formulation is for the PP-subproblem $ij, \forall ij \in L^v$. We borrow some notations from the RMP.

Given:

U_k : the integer value that denotes the residual CPU resources at node $k, k \in N^s$.

Variables:

$M_{mn,a}$: a Boolean variable which equals to 1 when physical link (m, n) is traversed and spectrum from the a th slot is used, otherwise 0.

A_k : a Boolean variable which equals to 1 when the virtual node i is mapped into physical node k , otherwise 0.

B_k : a Boolean variable which equals to 1 when the virtual node j is mapped into physical node k , otherwise 0.

T_a : a Boolean variable which equals to 1 if virtual link (i, j) is mapped to the substrate network and uses spectrum from the a th slot, otherwise 0.

Objective:

The objective function, i.e. reduced cost, of the PP-subproblem ij is derived by Lagrangian function, the derivation procedure is omitted for brevity.

$$\begin{aligned}
\text{Min} \quad & \sum_{mn \in L^s} \sum_{a \in F} \left(c_{mn} p_{ij} M_{mn,a} + \sum_{\substack{t=a-p_{ij}+1, \\ a-p_{ij}+2, \dots, a}} \theta_{mn,a} M_{mn,t} \right) \\
& + \sum_{k \in N^s} \left(\frac{c_k p_i + \gamma_k}{\text{Deg}(i)} + \text{Deg}(i) \alpha_{ij,k} - \sum_{t \in \text{oadj}(i)} \alpha_{it,k} - \sum_{t \in \text{iadj}(i)} \beta_{ti,k} \right) A_k \\
& + \sum_{k \in N^s} \left(\frac{c_k p_j + \gamma_k}{\text{Deg}(j)} + \text{Deg}(j) \beta_{ij,k} - \sum_{t \in \text{oadj}(j)} \alpha_{jt,k} - \sum_{t \in \text{iadj}(j)} \beta_{tj,k} \right) B_k - \pi_{ij}
\end{aligned} \quad (9)$$

where π_{ij} , $\alpha_{ij,k}$, $\beta_{ij,k}$, γ_k and $\theta_{mn,a}$ are the dual parameters of constraints (4), (5), (6), (7) and (8), respectively. The PP-subproblem ij is solved to obtain a PES and achieve the minimal reduced cost. If the value is non-negative, this PP-subproblem cannot find any PES to improve the objective value of the RMP. Otherwise, the derived PES is added into the RMP. If none of PP-subproblems derive negative reduced cost, the optimal solution of the original problem is obtained, and the iteration stops.

Constraints:

$$\sum_{a \in F} \sum_{m \in N^s} M_{mk,a} - \sum_{a \in F} \sum_{n \in N^s} M_{kn,a} = B_k - A_k, \quad \forall k \in N^s \quad (10)$$

Equation (10) ensures flow conservation of the physical path of the virtual link (i, j) . There are three cases according to the values of the right-hand side of the equation:

- 1) Equal to 0 (values of A_k and B_k are equal). According to Equation (13) below, a physical node can host at most one virtual node. Then, the values of A_k and B_k *cannot* be 1 simultaneously, so they must be equal to 0. Therefore, by the definitions of A_k and B_k , neither i or j is mapped to node k . Then, equation (10) ensures that the number of incoming used physical links is equal to the number of outgoing ones at node k .
- 2) Equal to 1 (B_k is equal to 1 and A_k is equal to 0). This means that the virtual node j is mapped to node k . Then, the number of incoming used physical links is larger than the number of outgoing ones by 1 at node k .
- 3) Equal to -1 (B_k is equal to 0 and A_k is equal to 1). This means that the virtual node i is mapped to node k . Then, the number of outgoing used physical links is larger than the number of incoming ones by 1 at node k .

$$\sum_{k \in N^s} A_k = 1 \quad (11)$$

$$\sum_{k \in N^s} B_k = 1 \quad (12)$$

$$A_k + B_k \leq 1, \quad \forall k \in N^s \quad (13)$$

$$p_i A_k + p_j B_k \leq U_k, \quad \forall k \in N^s \quad (14)$$

Equations (11), (12) and (13) ensure each virtual node must be mapped to one physical node, and a physical node can host one virtual node at most. Equation (14) ensures that there are sufficient CPU resources if the physical node is occupied by virtual node embedding.

$$\sum_{a \in F} T_a = 1 \quad (15)$$

$$\sum_{mn \in L^s} M_{mn,a} \geq T_a, \quad \forall a \in F \quad (16)$$

$$\sum_{mn \in L^s} M_{mn,a} \leq (|N^s| - 1)T_a, \quad \forall a \in F \quad (17)$$

$$p_{ij}M_{mn,a} \leq \sum_{t=a, \dots, a+p_{ij}-1} U_{mn,t}, \quad \forall mn \in L^s, a \in F \quad (18)$$

Equation (15) ensures that the PP-subproblem has one spectrum allocation. Equations (16) and (17) ensure that T_a has a non-zero value only when the virtual link embedding traverses physical link(s) and uses spectrum from a th slot. Equation (18) ensures that sufficient spectrum resources are allocated and the contiguity constraint is satisfied. Specifically, if virtual link (i, j) with the spectrum requirement p_{ij} uses spectrum from the a th slot of physical link (m, n) , the slots from a to $a + p_{ij} - 1$ in the physical link must be available.

4. Exact algorithm to solve the column generation model

To solve the column generation model, the RMP is first relaxed to a linear problem. Then, the branch-and-bound approach is used to derive the integer solutions of the RMP, and a tree topology will be generated during the procedure. Each node of the tree topology represents a problem for which a solution obtained by iterating between associated RMP and PP, and each of these nodes will henceforth be called a *tree node*.

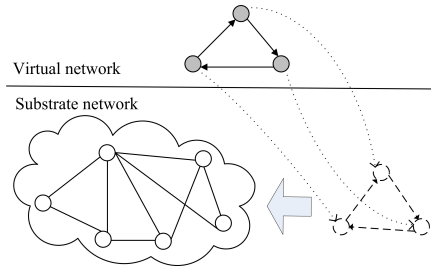


Fig. 3. An initial solution.

We propose the **Exact algorithm** shown in **Algorithm 1** to optimally solve the VNE problem of an arriving request. In the algorithm, TNL and UB are used to store tree nodes and denote the upper bound of the best solution already obtained, respectively. In line 2, an initial VNE solution is added into the relaxed RMP. An initial solution can be generated as shown in Fig. 3, where a small substrate network that is the same as the VN request is added, and disconnected to the original substrate network. The initial solution is obtained if the VN request is embedded within the small network. The embedding cost of the initial solution is set to a large value, which ensures the initial solution evolves to low cost ones. It is noted that the initial solution used in the algorithm is not a feasible solution for the VN request to be embedded in the original substrate network, because the initial solution comprises the added small substrate network that is disjointed from the original substrate network. If the initial solution is the same as the final solution then the VNE has failed and the VN request is blocked. In line 3, the RMP is relaxed to be a continuous linear programming, and in line 4, the root of the branch-and-bound tree is added into TNL . The loop from lines 5 to 12 is used to solve tree nodes and obtain the optimal solution. In line 6, a

Algorithm 1: Exact algorithm

Input: A VN request $G^v(N^v, L^v)$, and existing network resources.
Output: The optimal embedding solution Y^* , and updated network resources.
BEGIN:

- 1 $TNL \leftarrow \emptyset, UB \leftarrow \infty, Y^* \leftarrow \emptyset.$
- 2 Add the initial VNE solution into the RMP.
- 3 Relax the RMP to be a continuous linear programming.
- 4 Add the root of branch-and-bound tree into $TNL.$
- 5 **while** $TNL \neq \emptyset$ **do**
- 6 Pop a node from $TNL.$
- 7 Solve the relaxed RMP of the node to obtain solution Y and objective value $Z.$
- 8 Solve the PP of the node.
- 9 **If** a column with a negative reduced cost is found, add the column into the RMP, go to line 7.
- 10 **If** $Z \geq UB,$ go to line 5.
- 11 **If** Y is an integer solution, $Y^* \leftarrow Y$ and $UB \leftarrow Z.$
- 12 **If** Y is a fraction solution, branch and add two generated tree nodes into $TNL.$
- 13 **If** Y^* is not the initial solution, reserve resources for the VN request according to $Y^*.$
- 14 **Return.**
- END**

node is popped from TNL ; then, its RMP and PP are iteratively solved with a commercial solver, CPLEX [25], from lines 7 to 9 until the reduced cost is not negative. When iteration stops, the solution of the RMP is in one of three cases: 1) if Z is larger than the current best solution UB , this node will be discarded; 2) if Z is smaller than UB and Y is an integer solution, the current best solution Y^* and UB are updated; 3) otherwise two tree nodes are generated by branching and are added in TNL . After all nodes have been processed, Y^* is the optimal VNE solution. Finally, the network resources are allocated according to Y^* if it is not the initial solution. If the request cannot be accepted due to the lack of network resources, the output is the initial solution.

5. Heuristic algorithm

In the **Exact algorithm**, the optimal solution of the root in the branch-and-bound tree provides a lower bound of the final integer solution. Usually, this lower bound solution is fractional, and branching can be used to generate new tree nodes whose solutions are more likely integers. However, the number of generated tree nodes can be enormous, which increases the algorithm running time leading to scalability problems. In this section, a heuristic algorithm is proposed, and only the root of the branch-and-bound tree is solved to achieve a near optimal solution. In addition, solving the ILP problem of PP may be prohibitively time-consuming at line 8 of the **Exact algorithm**. We build an auxiliary graph in the heuristic algorithm to approximately solve the ILP problem.

At the beginning of the **Heuristic algorithm** shown in **Algorithm 2**, the initial solution is added and the RMP is continuously relaxed. The loop from line 4 to line 16 solves the root of the branch-and-bound tree, and the iteration between RMP and PP stops when the reduced cost becomes non-negative which means the optimal result is obtained. During the iterations, the best integer solutions already obtained is stored in Y^* , and it is updated at line 6 after the relaxed RMP solving if an integer solution is obtained. From line 7 to line 15, the PP is solved, and the first column with negative reduced cost is added to the RMP for next iteration.

In line 10, an auxiliary graph is constructed to find embedding solution for a virtual link and

Algorithm 2: Heuristic algorithm

Input: An arriving VN request $G^v(N^v, L^v)$, and existing network resources.
Output: An embedding solution Y^* , and updated network resources.
BEGIN:

- 1 $Y^* \leftarrow \emptyset$.
- 2 Add the initial VNE solution into the RMP.
- 3 Relax the RMP to be a continuous linear programming.
- 4 **do**
- 5 Solve the relaxed RMP to obtain solution Y .
- 6 **If** Y is an integer solution, $Y^* \leftarrow Y$.
- 7 **for** a PP-subproblem ij of the PP **do**
- 8 $rc \leftarrow 0$.
- 9 **for** $a = 1$ to $F - p_{ij} + 1$ **do**
- 10 Build the auxiliary graph with the spectrum band from slots a to $a + p_{ij} - 1$.
- 11 Derive the shortest path P from node i to node j .
- 12 **If** i and j are directly connected to V_m and V'_m ($\exists m \in N^s$) in the path P ,
respectively, delete a higher cost dummy link of P in the graph, go to line 11.
- 13 Let rc equals to the total cost of links traversed by P minus π_{ij} .
- 14 **If** $rc < 0$, break.
- 15 **If** $rc < 0$, break.
- 16 **If** $rc < 0$, the column (according to P) is added into the RMP.
- 17 **while** $rc < 0$
- 18 **If** Y is a fractional solution, replace Y^* if a better integer solution can be constructed from Y .
- 19 **Return.**
- 20 **END**

its two end nodes. For example, Figs. 4(a) and 4(b) are the substrate network and the auxiliary graph, respectively. In Fig. 4(b), dummy links (dash lines) between virtual nodes and physical nodes are connected and assigned values to denote the virtual node mapping costs. It is noted that only the physical nodes with sufficient residual CPU are connected. To satisfy the continuity constraint, the auxiliary graph is generated for a specific spectrum band. It is noted that only the physical links with sufficient continuous slots are added in the auxiliary graph. For example, for PP-subproblem ij in Fig. 4, each physical link (solid line) of auxiliary graph that is for spectrum assignment starting from the a th slot, should have available slots from a to $a + p_{ij} - 1$. In Fig. 4(b), the physical nodes are doubled, and each link is also duplicated as an original link from V_m to V_n to be a link from V_m to V'_n and a link from V'_m to V'_n . A shortest path algorithm is applied in line 11. If the derived path has a pair of nodes, e.g. V_m to V'_m , which means i and j are directly connected to a common physical node in the original graph (break constraint (13)), a higher cost dummy link in the path is deleted from the auxiliary graph. Then, the route is derived again.

The link cost of the auxiliary graph is set according to the cost terms in the objective function of the PP-subproblem ij in (9) that comprises virtual link (i, j) mapping term (with $M_{mn,a}$ and $M_{mn,t}$), virtual node i mapping term (with A_k), virtual node j mapping term (with B_k) and a dual parameter π_{ij} . These four terms are converted to link costs as follows.

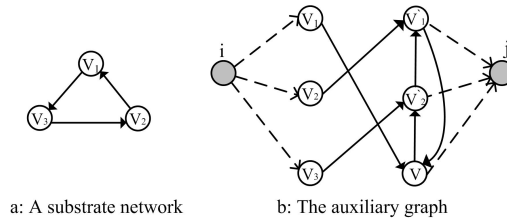


Fig. 4. Auxiliary graphs with available spectrum slots from a to $a + p_{ij} - 1$.

$$c'_{mn} = \begin{cases} \frac{c_n p_i + \gamma_n}{Deg(i)} + Deg(i) \alpha_{ij,n} - \sum_{t \in oadj(i)} \alpha_{it,n} \\ \quad - \sum_{t \in iadj(i)} \beta_{ti,n}, & m = i \\ \frac{c_m p_j + \gamma_m}{Deg(j)} + Deg(j) \beta_{ij,m} - \sum_{t \in oadj(j)} \alpha_{jt,m} \\ \quad - \sum_{t \in iadj(j)} \beta_{tj,m}, & n = j \\ c_{mn} p_{ij} + \sum_{t=a}^{a+p_{ij}-1} \theta_{mn,t}, & m \neq i, n \neq j, a \end{cases} \quad (19)$$

When the iterations stop, the optimal solution is obtained, and this optimal solution may be a fractional solution. In this algorithm, we use the rounding to the nearest integer to obtain an integer solution, if this integer solution is feasible and has a lower cost than Y^* , Y^* is updated in line 17. Finally, network resources are allocated in line 18.

6. Numerical results

In this section, we first provide numerical results to demonstrate the difference in performance of the two ways to solve the PP which is needed for column generation for a VNE request. Secondly, the performance of the VNE algorithms is compared in terms of blocking probability, embedding cost, virtual link embedding cost and virtual node embedding cost for two network topologies shown in Fig. 5.

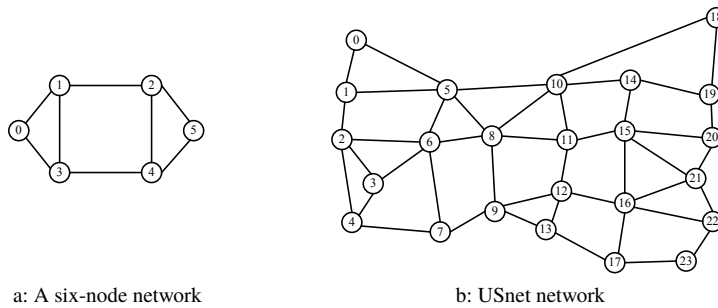


Fig. 5. Network topologies.

6.1. Two ways to solve PP

To show the differences between using ILP to solve PP (in the exact algorithm) and using the auxiliary graph to solve PP (in the heuristic algorithm), we give an example of solving one VN request in a six-node network (shown in Fig. 5(a)), and the parameter settings are shown in Table 1. The similar result has been observed in almost all other VN request solving, for the sake of brevity, they are not shown in this paper. Two PP solvings are indicated as heuristic PP and exact PP in Fig. 6 accordingly. The RMP objective function value and the running time are compared between exact PP (the root of branch-and-bound tree is considered) and heuristic PP in two figures, respectively. Figure 6(a) shows the RMP objective values are improved when the number of iterations increases. The exact PP and the heuristic PP overlap at the beginning. The reason is that both methods start from the same initial solution, and the RMP objective value does not change before a new feasible solution is found. Because the exact PP optimally solves PP and the heuristic PP use the auxiliary graph to obtain an approximate solution, two methods stop at different objective values in the figure, and the exact PP achieves a better value. It is noted that the two methods stop at different numbers of iterations, where the exact PP has fewer iterations than the heuristic PP. However, the execution time of the heuristic PP is about two orders of magnitude shorter than that of the exact PP (shown in Fig. 6(b)).

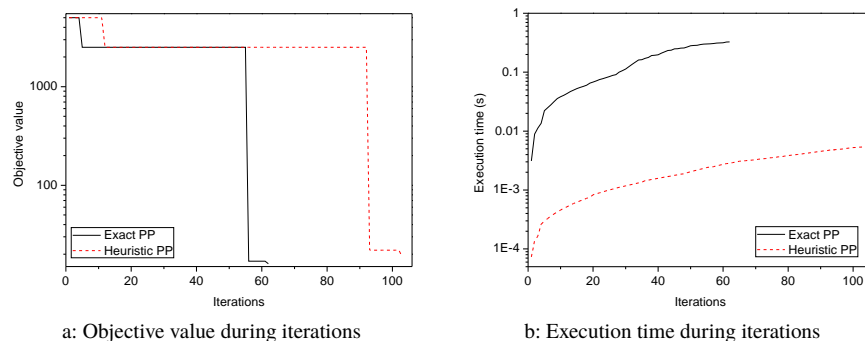


Fig. 6. Exact PP and heuristic PP comparison.

6.2. Comparison of VNE algorithms

The traffic scenario considered in this paper is dynamic, where VN requests arrive one-by-one over time, and both exact and heuristic algorithms solve one VN request at each time. From a long-time perspective, both algorithms only provide approximate solutions for the entire network operation period, which has practical interests but is much harder than the problem in this paper. In this subsection, we present numerical results for the exact and heuristic algorithms in the six-node network shown in Fig. 5(a), and will also provide results in a larger size practical network, USnet shown in Fig. 5(b), in the next subsection. Additionally, the heuristic-degree algorithm in our previous work [24] is selected for comparison, because it achieves the best performance than others in [24] when adapted to the problem in this paper.

In the simulations, VN request arrivals are assumed to be a Poisson process with arrival rate λ and their required service times are exponentially distributed with mean $1/\mu$. Then, the network load is given by λ/μ (Erlang). Table 1 shows the parameter settings in the six-node network and the Usnet network simulations. Error bars of 95% confidence intervals based on Student's t-distribution are provided for all results.

Figure 7 provides simulation results for cases where different network loads are considered. In Fig. 7(a), blocking probabilities are compared. Three algorithms have the same trend in the

Table 1. Parameter settings

Parameter	Six-node	Usnet
λ (arrival rate)	3.0 - 4.5	2.0 - 4.5
$1/\mu$ (mean of service time)	1 time-step	1 time-step
Node computation capacity	20	200
Link spectrum slots	20	50
CPU cost	1 cost-unit	1 cost-unit
Spectrum slot cost	unif(1,5) cost-unit	unif(1,5) cost-unit
Number of virtual nodes	unif(2,4)	unif(2,8)
Number of Virtual links	unif(1,6)	unif(1,10)
Computation requirement	unif(1,4)	unif(4,20)
Spectrum requirement	unif(1,6)	unif(2,10)
Number of requests in an experiment	10000	100000
Number of experiments	11	11

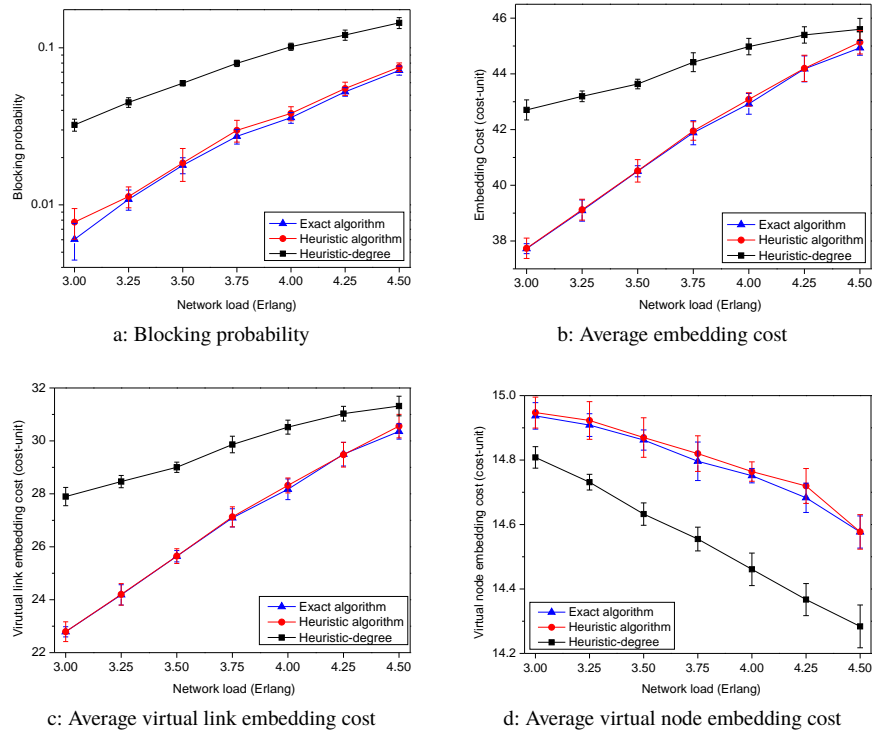


Fig. 7. Six-node network with variable network loads.

increase of the blocking probability as the network load increases, and the exact algorithm provides the lowest blocking probability, followed by the heuristic algorithm and the heuristic-degree. The value difference between the heuristic algorithm and the exact algorithm is quite small (0.2% average) as both algorithms are based on the column generation method. The heuristic algorithm has a lower blocking probability than the heuristic-degree by about 5%, and the first reasons is, the heuristic-degree was designed for the VNE modulation selection, not to the problem in this paper. Specifically, the heuristic-degree descending sorts the virtual links of a VN request

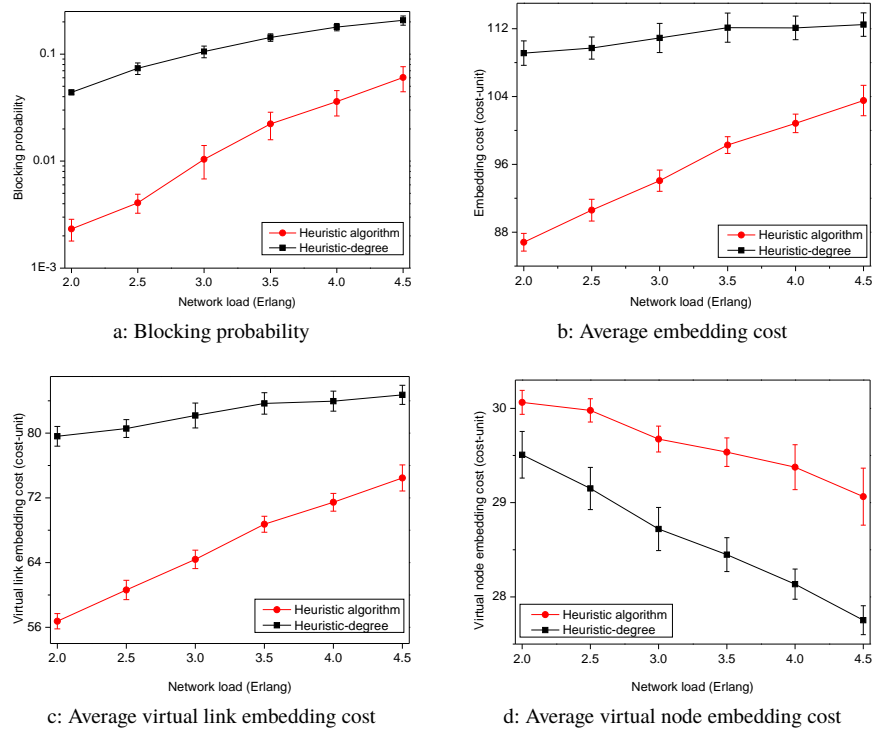


Fig. 8. USnet network with variable network loads.

according to the degrees of their virtual end nodes, and embeds virtual links one-by-one, which is to embed two adjacent virtual nodes close to each other to use short-length physical paths with high efficient modulation and reduce spectrum usages. However, when the heuristic-degree is adapted to the problem in this paper, short-length physical paths use a similar spectrum resource as long-length physical paths due to the single modulation, but may incur congestion problems because traffic is likely carried by spectrum resources within a limited zone. The second reason is that the heuristic algorithm applies more computations (using complex column generation) than the heuristic-degree, so it derives better VNE solutions.

In Fig. 7(b), the average embedding costs are compared. When the network load increases, the average embedding cost of VN requests increases. Three algorithms have the same orders as in Fig. 7(a), the exact algorithm has the lowest cost value, followed by the heuristic algorithm and the heuristic-degree, and the heuristic algorithm and the exact algorithm have almost the same value. The reason of the same order is that high embedding cost is usually caused by more resource consumptions, which leads to a high blocking probability. We also compare the average virtual link embedding cost (spectrum cost) and virtual node embedding cost (CPU cost) in Figs. 7(c) and 7(d), respectively. The virtual link embedding costs of three algorithms in Fig. 7(c) have the same orders as in the total embedding cost comparison in Fig. 7(b), and the heuristic algorithm and the exact algorithm have almost the same value. When the network load increases, longer paths may be used by virtual link embedding that leads to higher costs. However, in Fig. 7(d), the three algorithms have different orders as in Fig. 7(c) and the values decrease as the network load increases. The reason of the decrease is when the network load increases, more VN requests are blocked due to the lack of network resources, and the blocked requests likely require more computational resources than the non-blocked ones, then a lower

virtual node embedding cost is incurred. The similar reason can explain that the heuristic-degree has the lowest virtual node embedding cost in Fig. 7(d), specifically, the blocking probability of the algorithm is the highest and the requests with high computation requirements are more likely blocked. Even when the lowest virtual node embedding cost is obtained, the total embedding cost of the heuristic-degree is still the highest.

we also compare the performance of the heuristic algorithm and the heuristic-degree in a realistic size network, USnet. In Fig. 8(a), the heuristic algorithm has a lower blocking probability than the heuristic-degree by about one magnitude, and the embedding cost comparison in Fig. 8(b) indicates the reason of different blocking probabilities. Figures 8(c) and 8(d) show the virtual link embedding cost and virtual node embedding cost, respectively, and the same trend of lines in the figures can be observed as in the six-node network results. Also, the explanations provided for the case of the six-node networks are applicable here.

7. Conclusion

We have provided solutions for the VNE problem with dynamic traffic in flexi-grid networks that include an exact and a heuristic algorithms based on column generation. Both algorithms are designed for dynamic traffic scenarios where requests come one-by-one over times, and they aim to embed a VN request at minimal cost saving network resources. Numerical results have demonstrated very close (within 0.2% difference) blocking probability performance and average embedding cost of the two algorithms in a six-node network. For the larger USnet network, we have illustrated that the heuristic algorithm performs significantly better than the existing heuristic-degree algorithm used for a different VNE problem. These new solutions have the potential to speed up flexi-grid network virtualization, and to lead to future work where they can be extended to data center networks in which flexi-grid optical networks is likely to be deployed, meanwhile, a significant large number of the applications should be taken care of by the network resource management, which is the challenge issue in the future extension works.

Funding

National Natural Science Foundation of China (NSFC) (61401070, 61701079, 61671130, 61575126, 61671124); Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 11200417).