Service Function Path Provisioning With Topology Aggregation in Multi-Domain Optical Networks

Boyuan Yan, *Member, IEEE*, Yongli Zhao, *Senior Member, IEEE*, Xiaosong Yu, *Member, IEEE*, Yajie Li, Sabidur Rahman, *Member, IEEE*, Yongqi He, Xiangjun Xin, and Jie Zhang, *Member, IEEE*

Abstract—Traffic flows are often processed by a chain of Service Functions (SFs) (known as Service Function Chaining (SFC)) to satisfy service requirements. The deployed path for a SFC is called Service Function Path (SFP). SFs can be virtualized and migrated to datacenters, thanks to the evolution of Software Defined Network (SDN) and Network Function Virtualization (NFV). In such a scenario, provisioning of paths (i.e., SFPs) between virtualized network functions is an important problem. SFP provisioning becomes more complex in a multi-domain network topology. 'Topology aggregation' helps to create a single-domain view of such a network by abstracting multi-domain networks. However, traditional 'topology aggregation' methods are unable to abstract SF resources properly, which is required for SFP provisioning. In this paper, we propose an SFC-Oriented Topology Aggregation (SOTA) method to enable abstraction for SFs in multi-domain optical networks. This study explores the node and the link aggregation degree to evaluate information compression during the 'Topology aggregation' process. Additionally, we also propose a new data structure named wheel matrix and related operations to store routing information in the aggregated topology. Based on SOTA, we propose two cross-domain SFP provisioning algorithms named Ordered Anchor Selection (OAS) and *k*-paths OAS (K-OAS), and a benchmark named Global OAS (GOAS). Simulation results show that SOTA could aggregate large-scale multi-domain optical networks into a small network that contains only 6.9% of the nodes and 10.1% of the links. Both OAS and K-OAS can calculate SFPs efficiently and reduce blocking probability up to 52.10% compared to the benchmark.

Index Terms—Optical network, service function chain, service function path, topology aggregation.

I. INTRODUCTION

TOPOLOGY Aggregation (TA), also known as topology abstraction, is a critical technology for the management of large-scale multi-domain optical networks. TA provides an aggregated topology view for other domains to enable cross-domain service provisioning. A network domain can hide safety-sensitive information via the abstraction provided by TA, to improve confidentiality. In addition, TA helps to reduce the size of inter-domain communication which leads to enhancing scalability [1]. There are three basic TA methods proposed by Optical Internetworking Forum (OIF) [2] for Automatically Switched Optical Network (ASON) [3], as shown in Fig. 1.

Figure 1(a) shows the original topology of a single domain with four border nodes. Fig. 1(b) and Fig. 1(c) show Abstract node model and Abstract link model, respectively. Abstract node model is also known as simple-node aggregation [4], and abstract link model is also known as mesh aggregation. Figure 1(d) shows the combination of abstract link and node model, also known as star aggregation. Simple-node aggregation (i.e., node model) considers a domain as one aggregated node, which is very simple but would result in large topology information distortion. Mesh aggregation (i.e., link model) constructs a complete graph among the border nodes. The aggregated links represent routing cost between border node pairs in the domain. Mesh aggregation is more complex and would result in smaller topology information distortion. As Fig. 1(d) shows star aggregation introduces an aggregated node to show intra-domain routing information, compared to mesh aggregation. Apart from OIF, Internet Engineering Task Force (IETF) also proposed a similar TA scheme [5] in Abstraction and Control of Traffic engineering Network (ACTN) architecture. However, according to Ref. [6], most TA algorithms suffer from network distortion, and the key point is how to balance the trade-off between abstraction complexity and information distortion.

The research on TA mainly focuses on service provisioning and survivability in traditional scenarios. Wan *et al.* [7] proposed a bidirectional shuffle-net model to reduce the size of link state information of the TA procedure in ASON. Wang *et al.* [8] introduced a 'wavelength capable matrix'
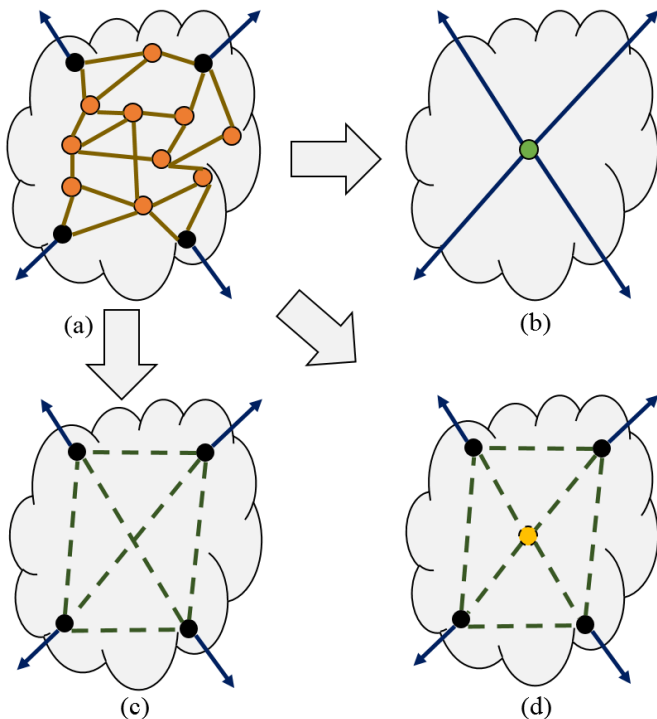
Fig. 1. Basic TA types (a) original topology without TA; (b) abstract node model; (c) abstract link model; (d) combination of abstract link and node model.

to represent wavelength convertor information from the physical network, which is helpful in path calculation. Casellas *et al.* [9] evaluated the scalability of mesh aggregation under hierarchical Path Computation Element (PCE) architecture. Cai *et al.* [10] proposed a novel TA algorithm that enables multiple Quality of Service (QoS) constraints in multi-domain optical networks. They also introduced 'wrongly rejected ratio' and 'crank back ratio' to evaluate TA performance. Gao *et al.* [1], [11], [13] proposed several TA algorithms for survivable multi-domain optical networks. Akyama *et al.* [14] discussed the reliability of simple-node aggregation and intra-domain protection. Zhang *et al.* [15] proposed several TA algorithms based on various protection requirements. All these studies mentioned above consider the abstraction for optical node and fiber link resources only, but not the new features in emerging scenario. Resources related to Service Functions (SFs) carried by optical networks are not considered yet. SF related resources provide various network services such as Virtual Private Gateway (VPN), Service-Level Agreement (SLA) monitoring, Authentication, Authorization and Accounting (AAA) server, load balancer, firewall, etc. Traditional SFs are integrated into middle-box equipment and deployed in the static or semi-static environment [16]. In such a static deployment, aggregation for SF resources was not necessary. In recent years, with the development of Software Defined Optical Networking (SDON) [17], Network Function Virtualization (NFV), [18, [19] and 5G [20], SFs can be virtualized as software and deployed dynamically on high-volume servers in data centers (DCs) [21]. DCs are usually connected to the transport optical node for high-speed and low-latency services. To simplify, SF capabilities are considered as the attributes on optical devices in optical networks.

In SF-related applications, a Service Function Chain (SFC) is defined by an ordered or partially ordered set of SFs that must be applied to data traffic traveling through the network [22]. Apart from SFs, the SFC also includes the bandwidth service requirement between SFs, source node and destination node. A Service Function Path (SFP) is a route to steer traffic to form a SFC [23]. For example, for the live stream application, the traffic flow is generally handled by several SFs including security clearance, video compression, data encryption, data decryption, video decompression, etc. When a SFP passes through multiple data centers across different optical network domains [24], the cross-domain SFP provisioning problem arises. There are some related research works about SFP dynamic provisioning problem. Our prior work [25] introduced the concept of 'anchor node' to represent the SF-enabled nodes. Sonkoly *et al.* [26] designed a joint cloud-network resource virtualization API to control SFCs over multi-domain networks. Sun *et al.* [27] proposed two methods for cross-domain SFC partitioning, and two heuristic algorithms to deploy the SFC. Especially, a feedback mechanism was proposed to improve the success ratio of SFC. Zhang *et al.* [28] presented a vertex-centric distributed orchestration for multi-domain networks. They also proposed a distributed computing algorithm to find feasible SFPs. In SFP provisioning works, only a few studies considered SFC-oriented TA in cross-domain SFP provisioning. Zhang *et al.* [29] proposed two simple TA algorithms: flat TA and auxiliary-edge-based TA. Both of the TA algorithms divides SFs into 'service functions' and 'switch functions' in networks composed of routers and switches. However, such a partition is too simple because there is a variety of SFs and more kinds of SFs are emerging. And in optical networks, the wavelength constraints [30] are not considered. In this paper, all of them are considered in different parts. The main contributions are: i) new TA algorithm named SFC-Oriented TA (SOTA) to consider each kind of SF independently on aggregated topology, ii) new data structure named wheel matrix to consider the constraints on intra-domain level, and adjust the trade-off between complexity and information distortion on demand, iii) two SFP provisioning algorithms named Ordered Anchor Selection (OAS) and k-paths OAS (K-OAS).

The rest of this paper is organized as follows. Section **II** describes the SFP provisioning problem in multi-domain optical networks. Section **III** introduces SOTA in detail. Section **IV** introduces 'wheel matrix' structure, which enables SOTA-based path computation. Section **V** proposes OAS and K-OAS, two wheel-matrix-based cross-domain SFP algorithms and a simple benchmark cross-domain SFP algorithm without TA named Global OAS (GOAS). Section **VI** evaluates the performance of proposed algorithms compared with the benchmark algorithm. Section **VII** concludes this paper.

II. PROBLEM STATEMENT

Figure 2 shows a universal process for cross-domain SFP provisioning in multi-domain optical networks. The virtual-aggregated topology for a single-domain physical network
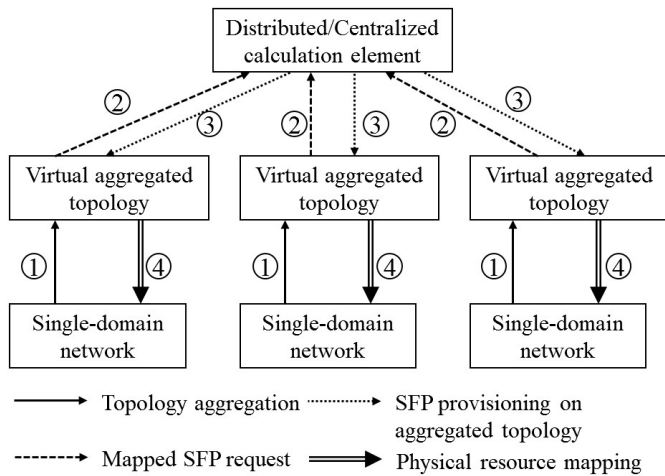
Fig. 2.   General process to calculate SFP under topology aggregation.

is calculated by every domain and remained in *calculation element* by following some aggregation updating strategy. The *calculation element* is a logical entity to allocate resource for multi-domain SFP, whose implementation can be centralized like cross-domain SDN controller or distributed like distributed PCE. The aggregation updating strategy controls aggregation synchronization between each domain and *calculation element*. Each domain decides its own sync strategy to keep the balance between synchronization costs and consistency requirements.

Firstly, when a cross-domain SFP request arrives, the request is mapped into virtual aggregated topology. Secondly, this information is sent to the distributed/centralized *calculation element*. Thirdly, calculation element calculates feasible cross-domain resource allocation on aggregated topology. And finally, SFP provisioned on aggregated topology are mapped into physical domains. Then, every single-domain network allocates the mapped resources to create an integrated cross-domain SFP.

In Fig. 2, TA is responsible to map and de-map network resources. Mapping from physical routing to aggregated link and de-mapping from aggregated link to physical routing is generally symmetrical process. But if the sync strategy is *loose*, sometimes mapping and de-mapping may be different. To make up for information deviation, we consider *strict* sync strategy and *symmetric* mapping to avoid those deviations. In the scenario of this manuscript, TA needs to hide the constraints inside an optical domain to expose common attributes for inter-operation in multi-domain heterogeneous networks. Also, when some capabilities (SF, etc.) are required from outside, TA should be able to provide a general way to abstract them. Therefore, we propose SOTA to handle these requirements, and wheel matrix on aggregated links to solve more capabilities that a single-domain could provide. Based on them, two cross-domain SFP provisioning algorithms are able to calculate feasible SFP on aggregated topology, and execute the resource allocation on physical networks.

III. SFC-ORIENTED TOPOLOGY AGGREGATION

In this paper, we propose an SFC-Oriented TA algorithm to aggregate optical network resources with SFs. In order to describe the algorithms accurately, the notations are listed as Table I. SOTA uses an irreversible information compression procedure which can be divided into two steps: node aggregation and link aggregation. Node aggregation maps physical nodes to virtual *aggregated nodes* and link aggregation builds virtual aggregated links among *aggregated nodes* according to available bandwidth, delay, SF resources, etc.

A. Node Aggregation

As Fig. 3(a) shows, the digits $\boxed{1}$, $\boxed{2}$, and $\boxed{3}$ represent the specific service function f_1, f_2, and f_3 respectively with one unit. For example, node n_1^1 in Fig. 3(b) are able to provide service function f_1 with 3 units, because there are three $\boxed{1}$ squares above this node. Fig. 3(a) also shows two SFCs $c_1 = [(f_1, 1, 2), (f_2, 1, 2), (f_3, 2, 2)]$ and $c_2 = [(f_3, 3, 2), (f_2, 3, 1)]$. Each SFC is an ordered triple list, and each triple indicates a routing segment requirement, including the kind of service function, the required SF resource number, and the bandwidth requirement to the next segment. For example, SFCc_1 is comprised of three service functions: f_1, f_2, and f_3. The resource requirements for f_1, f_2, and f_3 are 1, 1, and 2 units, respectively. Besides, bandwidth requirements between two adjacent SFs are 2 units. The bandwidth requirement from aggregated node enabling f_3 to the destination node is 2 units.

Figure 3(b) shows a three-domain optical network where Border Nodes (BNs) are connected to other domains and Physical Interior Nodes (PINs) provide available bandwidth for intra-domain routing. BNs act as gateways for inter-domain information exchange, making it harder to hide BN information from outsiders. The number of BNs in a domain is usually smaller than that of PINs. PINs' information is usually confidential for operators, and should not be exposed outside.

In order to abstract SF resources, we aggregate all SF-enabled Nodes (SNs) into Aggregated SF-enabled Nodes (ASNs). Figure 3(c) shows the ASNs $\{v_1, v_2, v_6, v_7, v_{11}, v_{12}\}$ generated from SNs in Fig. 3 (b). The numbers shown on over the SNs (in Fig. 3(b)) and ASNs (in Fig. 3(c)) indicate the corresponding SF resources those nodes can provide. After aggregation, each type of SF resource is virtualized as an ASN. For example, in domain 1, node v_2 aggregates f_1 resources from n_1^1 and n_3^1 (i.e., $map(v_2) = [n_1^1, n_3^1]$). Similarly, v_1 aggregates f_2 resources from n_4^1 and n_6^1 (i.e., $map(v_1) = [n_4^1, n_6^1]$). This allows the proposed *node aggregation* method to abstract both SF and optical network resources.

B. Link Aggregation

After node aggregation, BNs are reserved while SF resources are aggregated as virtual nodes. The next step is to build connections between these nodes. Link aggregation only decides whether two aggregated nodes have a directed connection. Link attributes depend on the mapping method between the aggregated link to multiple routing paths in physical networks. In Section IV, we propose wheel matrix and related operations to support such a mapping. In link aggregation method, all intra-domain links are discarded and new aggregated links are created to connect ASNs and
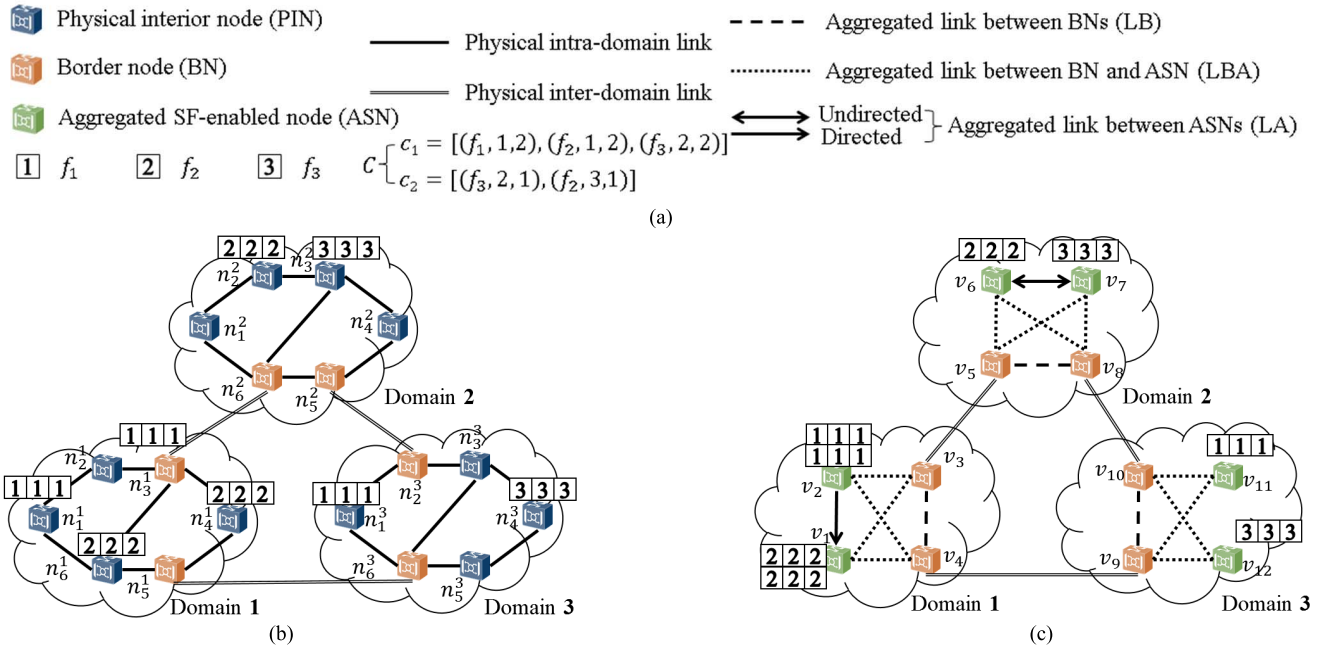
Fig. 3. SFC-oriented TA algorithm (a) legends; (b) original physical SF-enabled topology; (c) virtual SF-enabled topology after SOTA.

BNs. Aggregated links could be divided into three types (Fig. 3(c)): Aggregated Link between BNs (LB), Aggregated Link between BN and ASN (LBA), and Aggregated Link between ASNs (LA). There must be a LB between any two BNs, in order to provide routing that only passes through this domain without SF allocation. For example, domain 1 in Fig. 3(c) contains only one LB between v_3 and v_4, i.e., n_3^1 and n_5^1 in Fig. 3(b). LBA represents connectivity from domain border to the node that carries SF resources. For example, in Fig. 3(c), there are 4 LBAs connecting BN set $\{v_3, v_4\}$ and ASN set $\{v_1, v_2\}$.

The LAs are used to connect ASNs to create cascaded traffic flow among different ASNs in a domain. The LAs are based on the adjacent relationships between SFs, which is extracted from SFC set. For example, the two SFCs c_1 and c_2 contain three directed connections between SFs (i.e., the connection from f_1 to f_2, the connection from f_2 to f_3, and the connection from f_3 to f_2). Based on these directed connections, domain 1 introduces a directed LA from v_2 to v_1. Similarly, domain 2 contains an undirected LA between v_6 and v_7.

C. Aggregation Degree

Based on node aggregation and link aggregation, we also propose node aggregation degree (D_N) and link aggregation degree (D_L) to evaluate the degree of information compression. We use aggregation constraints (1-3) to explain the relationships between nodes and links. Based on these constraints, D_N and D_L are defined as (4-5) show. D_N considers N_B as both numerator and denominator, so its scope is (0, 1]. D_L skips border links as they connect two domains and do not belong to any domain. The smaller D_N and D_L are, the deeper

information compression will be.

$$L_B = \frac{N_B \cdot (N_B - 1)}{2} \quad (1)$$

$$L_{BA} = N_B \cdot V_S \quad (2)$$

$$L_A \leq \frac{V_s \cdot (V_s - 1)}{2} \quad (3)$$

$$D_N = \frac{N_B + V_S}{N_B + N_I} \quad (4)$$

$$D_L = \frac{L_B + L_A + L_{BA}}{E_I}$$
$$= \frac{2L_A + N_B \cdot (2V_S + N_B - 1)}{2E_I} \quad (5)$$

IV. WHEEL MATRIX AND RELEVANT SOTA OPERATIONS

After node aggregation and link aggregation, the aggregated node may be mapped to a list of multiple physical nodes, and the aggregated link may be mapped to much more source-destination pairs on physical networks. In order to include source-mapping and destination-mapping information in aggregated nodes, we propose a new data structure named *wheel matrix*. This data structure enables information mapping between physical nodes and aggregated nodes. Equation (6) shows the *wheel matrix* M_i^j for two aggregated nodes v_i and v_j. a and b are the number of mapped physical nodes from aggregate node v_i and v_j, respectively.

$$M_i^j = \begin{bmatrix} m_{11} & \cdots & m_{1b} \\ \vdots & \ddots & \vdots \\ m_{a1} & \cdots & m_{ab} \end{bmatrix}_{a \times b},$$
$$a = |map(v_i)|, b = |map(v_j)| \quad (6)$$

TABLE I
SYMBOL DEFINITION

Symbol	Meaning
N	Node set of a multi-domain optical network.
V	Aggregated node set of multi-domain virtual aggregated topologies.
C	Set of SFC carried in a multi-domain optical network.
S	Set of SF supported by a multi-domain network.
n_i^j	The *i*-th node of domain *j*.
$c = [(f, r_f, b_f) \mid f \in S]$	A SFC that contains $\mid c \mid$ tuples of service function f, related resource requirement r_f, and bandwidth requirement b_f before the process of this service function in order.
N_I	Number of intra-domain physical nodes in a single domain.
N_B	Number of border physical nodes in a single domain.
V_S	Number of aggregated SF-enabled nodes on aggregated topology in a single aggregated domain.
E_I	Intra-domain physical link number in a single domain.
L_B	Aggregated link number between aggregated border nodes in a single aggregated domain.
L_A	Aggregated link number between aggregated SF-enabled nodes in a single aggregated domain.
L_{BA}	Aggregated link number between aggregated SF-enabled node and border node in a single aggregated domain.
$map(v \mid v \in V)$	Physical node list mapped from aggregated node v.
$demap(n_i^j \mid n_i^j \in N)$	Aggregated node list mapped from physical node n_i^j.
$req(c, f \mid c \in C, f_i \in S)$	Resource requirement of service function f in SFC c.
$adj(v \mid v \in V)$	Adjacent node set of aggregated nodes v.
$cap(n, f \mid n \in N, f \in S)$	Capability to support service function f of physical node v.
$R(s, d, b, c \mid s, d \in N, s \neq d, c \in C)$	SFP request where s is the source node, and d is the destination node. b is bandwidth requirement from last anchor node to d and c is required SFC.

M_i^j is a matrix containing the connection information from aggregated node v_i to v_j. For example, m_{pq} is the element of M_i^j, located at the p-th row and the q-th column. m_{pq} means the routing from p-th mapped node of $map(v_i)$ to the q-th mapped node $map(v_j)$ in the physical network. m_{pq} is a one-dimensional array and each element of it represents a link attribute. There are totally three kinds of link attributes in m_{pq}: additive attributes α_{pq} (transmission delay, etc.), bottleneck attributes β_{pq} (available bandwidth, etc.), and multiplicative attributes γ_{pq} (bit error rate, etc.), as (7) shows.

$$m_{pq} = [\underbrace{\alpha_{pq}^1, \alpha_{pq}^2, \ldots,}_{additive} \underbrace{\beta_{pq}^1, \beta_{pq}^2, \ldots,}_{bottleneck} \underbrace{\gamma_{pq}^1, \gamma_{pq}^2, \ldots}_{multiplicative}] \quad (7)$$

If only one of v_i and v_j is BN, then M_i^j is a $1 \times b$ row vector or a $a \times 1$ column vector, because BN could be mapped to one border node on the physical network. If both v_i and v_j are BNs, M_i^j is a 1×1 matrix. If the source and destination physical node are the same in m_{pq} (For example, if the border node is also a SF-enabled node, it will be aggregated as a BN and into a ASN on the aggregated network.), the additive parameter α_{pq} is 0, the bottleneck attribute β_{pq} is $+\infty$, and the multiplicative attribute γ_{pq} is 1. m_{pq} can also be used to calculate a metric to evaluate the performance of the physical routing from p to q. The best metric of a *wheel matrix* is the minimum or maximum value of all metrics in a *wheel matrix*. Besides, the definition of wheel matrix is to support roll operation and continuous roll operation to enable SFP calculation in an aggregated topology.

A. Roll Operation

Classical routing algorithms (*Dijkstra's* algorithm, etc.) could be divided into depth-first and breadth-first algorithms. In almost all routing algorithms, the traversal for nodes and links is fundamental and mandatory. Wheel matrix expands the link attribute as a matrix. And the traversal rule from one node to its adjacent node on the wheel matrix is different from traditional methods. A new traversal rule on aggregated topology named roll operation (represented by $\oplus$) is defined.

$$M_i^j|_{a \times b} \oplus N_j^k|_{b \times c}$$

$$= \begin{bmatrix} m_{11} & \cdots & m_{1b} \\ \vdots & \ddots & \vdots \\ m_{a1} & \cdots & m_{ab} \end{bmatrix}_{a \times b} \oplus \begin{bmatrix} n_{11} & \cdots & n_{1c} \\ \vdots & \ddots & \vdots \\ n_{b1} & \cdots & n_{bc} \end{bmatrix}_{b \times c}$$

$$= \begin{bmatrix} p_{11} & \cdots & p_{1c} \\ \vdots & \ddots & \vdots \\ p_{a1} & \cdots & p_{ac} \end{bmatrix}_{a \times c} = P_i^k|_{a \times c} \quad (8)$$

$$p_{xy} = best(\{mn_{xzy} \mid z \in [1, b]\}), \quad x \in [1, a], y \in [1, c] \quad (9)$$

Equation (8) shows the calculation of aggregated routing $v_i \rightarrow v_j \rightarrow v_k$. The order of resulting matrix P_i^k by roll
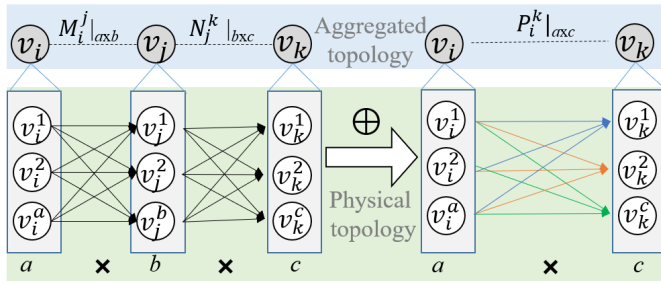
Fig. 4. Demonstration of roll operation.

operation is the same as that of normal matrix multiplication, i.e., the row size of P_i^k is the same as M_i^j, the column size of P_i^k is the same as N_j^k, and the column size of M_i^j must be the same as the row size of N_j^k. Roll operation is a compression process that skips poorer routing options from v_i to v_k, as Fig. 4 shows. In Fig. 4, each connection between two nodes represents an element of wheel matrix. For example, the connection from v_i^1 to v_j^2 means the element m_1^2 of M_i^j. For each physical source-destination pair (x, y) where $x \in map(v_i)$ and $y \in map(v_k)$, there are total b routing options, as the intermediate node v_j can be mapped to b physical nodes. So, in one roll operation, there are $a \times b \times c$ routing options to be considered and compressed to $a \times c$ better options. While calculating each element of P_i^k, the compression procedure is completed by a function named $best(.)$, as (9) shows. $best()$ is a filter function to select the best one from b choices of physical routing segments by following a specific rule. Such a rule could be defined by the network operator or developer.

The input parameter of $best()$ includes b sets of link attributes, each element m_{xzy} means the combination of two physical routing paths $x \to z$ and $z \to y$. Equation (10) shows the operator $\odot$, proposed to calculate the combination of link attributes in roll operation. This new operator is named Operator for Multiple Attributes (OMA). Equations (11-13) show different operational rules for three types of attributes.

$$mn_{xzy} = m_{xz} \odot n_{zy}$$
$$= [\underbrace{\alpha_{xzy}^1, \alpha_{xzy}^2, \dots,}_{additive} \underbrace{\beta_{xzy}^1, \beta_{xzy}^2, \dots,}_{bottleneck} \underbrace{\gamma_{xzy}^1, \gamma_{xzy}^2, \dots}_{multiplicative}]$$

$$(10)$$

$$\alpha_{xzy} = \alpha_{xz} + \alpha_{zy} \qquad (11)$$
$$\beta_{xzy} = \min(m_{xz}, m_{zy}) \qquad (12)$$
$$\gamma_{xzy} = \gamma_{xz} \cdot \gamma_{zy} \qquad (13)$$

For every roll operation of M_i^j and N_j^k, there should be $a \times b \times c$ OMA operations, and $a \times c$ comparison by using method. The time complexity of OMA operation is a constant that depends on the size of link attributes. And the time complexity of $best()$ depends on its implementation. Besides, how many times of roll operations for every cross-domain SFP calculation depends on the aggregated SFP algorithm.

The operator $\oplus$ is not commutative, because $M_i^j \oplus N_j^k$ and $M_j^k \oplus N_i^j$ mean opposite direction. Operator $\oplus$ does not obey the associative law as well. Equation (14) shows the inequation resulting from $(a \times b \times c + a \times c \times d)$ times of OMA on the left and $(b \times c \times d + a \times b \times d)$ times of OMA on the right.

As for $\odot$, it is easy to prove that OMA is not commutative, but obeys the associative law.

$$\underbrace{M_i^j|_{a \times b} \oplus N_j^k|_{b \times c} \oplus O_k^l|_{c \times d}}_{(V_i \to V_j \to V_k) \to V_l} \neq \underbrace{M_i^j|_{a \times b} \oplus (N_j^k|_{b \times c} \oplus O_k^l|_{c \times d})}_{V_i \to (V_j \to V_k \to V_l)}$$

$$(14)$$

B. Continuous Roll Operation

As mentioned above, $\oplus$ is a binary operator to combine only two routing segments on aggregated topology. For multi-segment routing calculation, roll operation is often unable to find good routing paths, because some options performing well globally are discarded in local roll operations (due to poor performance). In order to calculate better routing paths, a new multivariate operator named Continuous Roll Operation (CRO) (represented symbolically by $\circ()$), is proposed.

Equation (15) shows ternary CRO to calculate routing $v_i \to v_j \to v_k \to v_l$. Equation (16) shows q_{xy} from $b \times c$ choices according to method $select(.)$. As an extension of roll operation, it is easy to prove that CRO does not obey commutative law and associative law too.

$$\circ(M_i^j|_{a \times b}, N_j^k|_{b \times c}, O_k^l|_{c \times d})$$
$$= \begin{bmatrix} q_{11} & \cdots & q_{1d} \\ \vdots & \ddots & \vdots \\ q_{a1} & \cdots & q_{ad} \end{bmatrix} = Q_i^l|_{a \times d} \qquad (15)$$
$$q_{xy} = select(\{m_{xw} \odot n_{wz} \odot o_{zy} | w \in [1, b], z \in [1, c]\}),$$
$$x \in [1, a], y \in [1, d] \qquad (16)$$

In general, wheel matrix is a quietly flexible way to control complexity, security, and other key factors. One the one hand, the single-domain operator is able to hide some nodes, links, or even routing paths by adjusting the content of wheel matrixes. Especially, in order to reduce the complexity, the operator could also make a dynamic policy to expose different parts of its own domains at different time. On the other hand, wheel matrix may introduce some flags to customize the roll operation to accelerate the calculation or implement a particular function. For example, the flag of Shared Risk Link Group could be added as link attribute to provide link-disjoint SFP.

V. CROSS-DOMAIN ROUTING ALGORITHMS

Based on the *wheel matrix* and its operations defined above, we propose Ordered Anchor Selection (OAS) algorithm to support cross-domain SFC. We also propose k-paths OAS (K-OAS) to improve the performance of OAS by introducing more calculating. Then, a benchmark algorithm named Global OAS (GOAS) is introduced to compare the performance of OAS and K-OAS. Both OAS and K-OAS are based on the aggregated topology. And the aggregated topology is synchronized by all single domains loosely or strictly. While building the wheel matrixes of aggregated topology, the aggregated link attributes are calculated by proper RSA algorithms in each domain. In this paper, the classic shortest-path and first-fit RSA algorithm is used to calculate the aggregated topology for OAS and K-OAS.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

YAN *et al.*: SERVICE FUNCTION PATH PROVISIONING WITH TOPOLOGY AGGREGATION IN MULTI-DOMAIN OPTICAL NETWORKS 7

Algorithm 1 Pseudo-Code for OAS Algorithm

Input: Cross-domain SFP request *R(s, d, b, c)*
1 **if** $\forall f \in S, \exists n \in N, s.t.cap(n, f) \geq req(c, f)$ **then**
2 $\quad v_s \leftarrow demap(s), v_d \leftarrow demap(d)$;
3 $\quad v \leftarrow v_s, M \leftarrow M_{v_s}^v$;
4 $\quad$ Append tuple *(null, null, b)* to *c*;
5 $\quad$ **forall** *(f,r,b)* $\in c$ *in order* **do**
6 $\quad\quad$ *H* is initialized as an empty heap;
7 $\quad\quad$ Add *v* into *H*;
8 $\quad\quad T \leftarrow$
$\quad\quad \{t | \exists n \in \ map(t), t \in V, \ s.t.cap(n, f) \geq$
$\quad\quad req(c,f)\}$;
9 $\quad\quad x \leftarrow v$;
10 $\quad\quad$ **while** $x \notin T$ **do**
11 $\quad\quad\quad$ **forall** $k \in adj(x)$ **do**
12 $\quad\quad\quad\quad$ **if** *(k is a BN, or k in T) &&(k has't*
$\quad\quad\quad\quad$ *been visited in* **while loop***)* **then**
13 $\quad\quad\quad\quad\quad M_v^k \leftarrow M_v^x \oplus M_x^k$;
14 $\quad\quad\quad\quad\quad$ Update *H* according to M_v^k;
15 $\quad\quad\quad$ **if** $\forall h \in H M_v^h,$ *is unreachable* **then**
16 $\quad\quad\quad\quad$ *R(s, d, b, c)* is blocked;
17 $\quad\quad\quad x \leftarrow$ node popped by *H*;
18 $\quad\quad M \leftarrow M_{v_s}^v \oplus M_v^x, v \leftarrow x$;
19 $\quad$ Allocate bandwidth and SF resource through *M*;
20 $\quad$ **if** *Allocation failed* **then**
21 $\quad\quad$ *R(s, d, b, c)* is blocked.
22 **else**
23 $\quad$ *R(s, d, b, c)* is blocked.

Algorithm 2 Pseudo-Code for K-OAS Algorithm

Input: Cross-domain SFP request *R(s, d, b, c)*, *k*
1 **if** $\forall f \in S, \exists n \in N, s.t.cap(n, f) \geq req(c, f)$ **then**
2 $\quad v_s \leftarrow demap(s), v_d \leftarrow demap(d)$;
3 $\quad W \leftarrow [v_s], K \leftarrow [M_{v_s}^v]$;
4 $\quad$ Append tuple *(null, null, b)* to *c*;
5 $\quad$ **forall** *(f,r,b)* $\in c$ *in order* **do**
6 $\quad\quad T \leftarrow$
$\quad\quad \{t | \exists n \in \ map(t), t \in V, \ s.t.cap(n, f) \geq$
$\quad\quad req(c,f)\}$;
7 $\quad\quad$ **forall** $i \in \{1 \dots |K|\}$ *in order* **do**
8 $\quad\quad\quad v \leftarrow W[i], M \leftarrow K[i], x \leftarrow v$;
9 $\quad\quad\quad$ *H* is initialized as an empty heap;
10 $\quad\quad\quad$ Add *v* into *H*;
11 $\quad\quad\quad$ **while** $x \notin T$ **do**
12 $\quad\quad\quad\quad$ **FOR loop**from Step 11 to 14 in Algorithm **I;**
13 $\quad\quad\quad\quad$ **if** $\forall h \in H, M_v^h$ *is unreachable* **then**
14 $\quad\quad\quad\quad\quad$ Remove $i - th$ element of *W* and
$\quad\quad\quad\quad\quad$ *M*,**break**;
15 $\quad\quad\quad\quad x \leftarrow$ node popped by *H*;
16 $\quad\quad\quad M \leftarrow M_{v_s}^v \oplus M_v^x, v \leftarrow x$;
17 $\quad$ **if** $|K| = 0$ **then**
18 $\quad\quad$ *R(s, d, b, c)* is blocked.
19 $\quad$ **if** $|K| < k$ **then**
20 $\quad\quad p \leftarrow min\{k - |K|, |T| - 1\}$;
21 $\quad\quad$ Copy *W[0]* and *K[0]* to append to *W* and *K*
$\quad\quad$ separately for *p* times;
22 $\quad\quad$ **forall** $i \in [|K| - p + 1, |k|]$ *in order* **do**
23 $\quad\quad\quad$ Repeat Step 8-16 in this algorithm;
24 $\quad$ Select a best solution from $|K|$ options, and allocate
$\quad$ resource;
25 $\quad$ **if** *Allocation failed* **then**
26 $\quad\quad$ *R(s, d, b, c)* is blocked.
27 **else**
28 $\quad$ *R(s, d, b, c)* is blocked.

A. Ordered Anchor Selection (OAS) Algorithm

Algorithm I show the pseudo-code for OAS algorithm. Upon arrival of a cross-domain SFP request, the first **IF** block checks whether there are enough SF resources on the whole network at Step 1. If not, the request should be blocked directly at Step 23. If there are enough resources, the new request is mapped on aggregated topology at Steps 2-4. Then, The **FOR** loop between Steps 5-18 calculates the SFP allocation. After that, the allocation is executed on the physical multi-domain network. Step 20 checks whether the allocation fails because of information distortion while aggregating topology.

At Step 6, an empty Fibonacci heap *H* is initialized. Fibonacci heap is a special data structure including a set of the tree. This heap can achieve efficient priority queue following some ordering rule and is usually used in routing calculation. In OSA algorithm, *H* contains nodes in aggregated topology, and pops the node with minimal best metric, which is calculated from wheel matrix. At Step 7, the aggregated source node is added into *H* firstly. At Step 8, set *T* contains available ASNs whose mapped physical nodes could satisfy SF requirements. The **WHILE** loop from Steps 10-17 seeks an available routing from *v* to its next ASN on the aggregated topology. The **FOR** loop from Steps 11-14 traverses all adjacent nodes of node *x* that are visited in this iteration of WHILE loop. After visiting a BN/ASN, the wheel matrix from

v to the visited one is calculated and *H* will be updated. Step 15 checks if all nodes in *H* are unreachable. If yes, *R* is blocked. And if not, *H* pops the node with minimal *best metric* from the wheel matrix (as *x*).

At Step 12, the conditional statement contains a limit of "*k hasn't been visited*". This limit reduces the time complexity because every node can be reached only once. It means aggregated topology must be a graph without negative metric in any of the wheel matrixes.

OAS uses roll operation to calculate wheel matrixes of *H* at Step 13. As mentioned earlier, CRO can search more conditions than any combination of roll operations. However, in case of non-constrained routing calculation, the performance of CRO is the same as the performance of roll operation, but CRO costs more. Because constraints make roll operation non-linear, and CRO is equal to multiple continuous roll operations without constraints. Equation (17) and (18) show the relationship between roll operation and CRO in non-constrained

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

8

IEEE/ACM TRANSACTIONS ON NETWORKING

Algorithm 3 GOAS Algorithm

Input: Cross-domain SFP request $R, (s, d, b, c)$;

1   $A \leftarrow []$;
2   **if** $\forall f \in S, \exists n \in N, s.t. cap(n, f) \geq req(c, f)$ **then**
3     $n_c \leftarrow s$;
4   **forall** $(f, r, b) \in c$ $in\,order$ **do**
5       $T \leftarrow$
       $\{t | \exists n \in map(t), t \in V, s.t. cap(n, f) \geq req(c, f)\}$;
6   **forall** $t \in T$ **do**
7   Calculate routing and bandwidth segment allocation from n_c to t;
8   select optimal segment allocation from $|T|$ options, record the next anchor node n_n;
9       $n_c \leftarrow n_n$;
10   Calculate last segment routing to destination d;
11   Allocate bandwidth and SF resource on physical network;
12   **else**
13    $R(s, d, b, c)$ is blocked.

---

routing calculation. For a k-hop routing wheel matrix $M_1^k = \circ(M_2^3, M_3^3, \ldots, M_{k-2}^{k-1}, M_{k-1}^k)$, each element $m_{\alpha\beta}$ represents the best k-hop routing $(n_1, n_2, \ldots, n_{k-1}, n_k)|_{\alpha\beta}$ of all feasible routing from n_1 to n_k. Hence, the wheel matrix $M_1^{k-1} = \circ(M_1^2, M_2^3, \ldots, M_{k-2}^{k-1})$ contains the element that represents the best route.

$$M_1^k = M_1^{k-1} \oplus M_{k-1}^k \qquad (17)$$
$$\circ(M_2^3, M_3^3, \ldots, M_{k-2}^{k-1}, M_{k-1}^k)$$
$$= M_1^2 \oplus M_2^3 \oplus \ldots \oplus M_{k-2}^{k-1} \oplus M_{k-1}^k \qquad (18)$$

B. K-OAS Algorithm

Compared with OAS, K-OAS is designed to search more available paths on aggregated topology. K-OAS tries to find k feasible paths on aggregated topology to improve the performance. Algorithm II shows the pseudo-code for K-OAS.

In Step 3, K-OAS uses ordered list W and K to save ASNs and wheel matrixes from source s. In Steps 6-23, the **FOR** loop aims to find k feasible SFP allocation methods. Then in Step 24, we choose the best solution from calculated $|K|$ options and execute it. If the execution fails, it means this request is blocked.

In Steps 8-16, K-OAS traverses each uncompleted candidate SFP, and calculate the next ASN for it. This procedure is similar to the calculation of OAS. If the segment calculation fails, Step 14 removes the related elements in W and K. If K becomes an empty list (indicating no available routing), the SFP request is blocked at Step 18. If the size of K is less than k (i.e., the number of available paths from source to the destination is less than k), then K-OAS would fork new branches by copying the first element of K (Steps 20-21). Step 23 calculates different segment paths. Finally, K-OAS allocates SF and FS resources according to list K at Step 24.
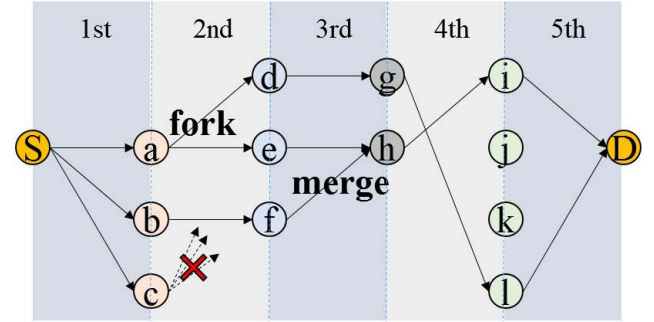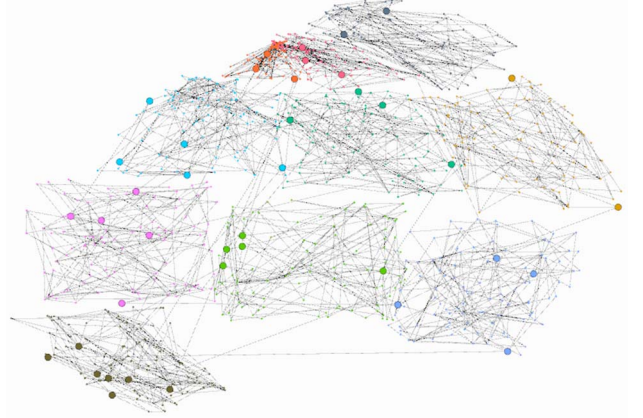

Fig. 5.   Brief demonstration of K-SOAS where $k = 3$.

Fig. 6.   Demonstration of 9-domain optical network.

In order to explain how to keep k candidate paths while calculating the SFP, Fig. 5 shows a brief demonstration with $k = 3$. S is the source and D is the destination of a SFP request on the aggregated topology. The SFP requires four kinds of SFs. $\{a, b, c\}$, $\{d, e, f\}$, $\{g, h\}$, and $\{i, j, k, l\}$ in the same color represent the four kinds of candidate ASNs with same type of SFs respectively. S, D and four sets of ASNs split the SFP into 5 routing segments. In the first segment, node a, b, and c are selected as possible k available paths, as the Steps 8-16 do. However, in the second segment, if node c fails to reach any next ASN of $\{d, e, f\}$, i.e., Step 14 is executed. Then the **IF** block of Steps 20-23 will be executed to add new potential SFP into $|K|$. Therefore, the path $S \rightarrow a$ is forks into $S \rightarrow a \rightarrow d$ and $S \rightarrow a \rightarrow e$ in Fig. 5 to update the total number of the paths to k. In the third segment, if the next ASNs for the path $S \rightarrow a \rightarrow e$ and the path $S \rightarrow b \rightarrow f$ are the same ASN h. These two paths merge into node h on aggregated topology. However, there is no merging operation because ASN h may be mapped into different physical nodes in different paths. Finally, k available paths on aggregated topology are $S \rightarrow a \rightarrow d \rightarrow g \rightarrow l \rightarrow D$, $S \rightarrow a \rightarrow e \rightarrow h \rightarrow i \rightarrow D$, and $S \rightarrow b \rightarrow f \rightarrow h \rightarrow i \rightarrow D$. Compared to OAS, the most important point of K-OAS is that K-OAS tries to remain k feasible solutions for each roll operation on the aggregated topology. The *fork* and *merge* operations handle the problem where the size of feasible solutions is less or more than k. It's necessary to clarify that when k=1, K-OAS will degrade to OAS. In this case, both fork and merge operations are useless. However, when k>1, K-OAS becomes more flexible and scalable.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

YAN *et al.*: SERVICE FUNCTION PATH PROVISIONING WITH TOPOLOGY AGGREGATION IN MULTI-DOMAIN OPTICAL NETWORKS 9
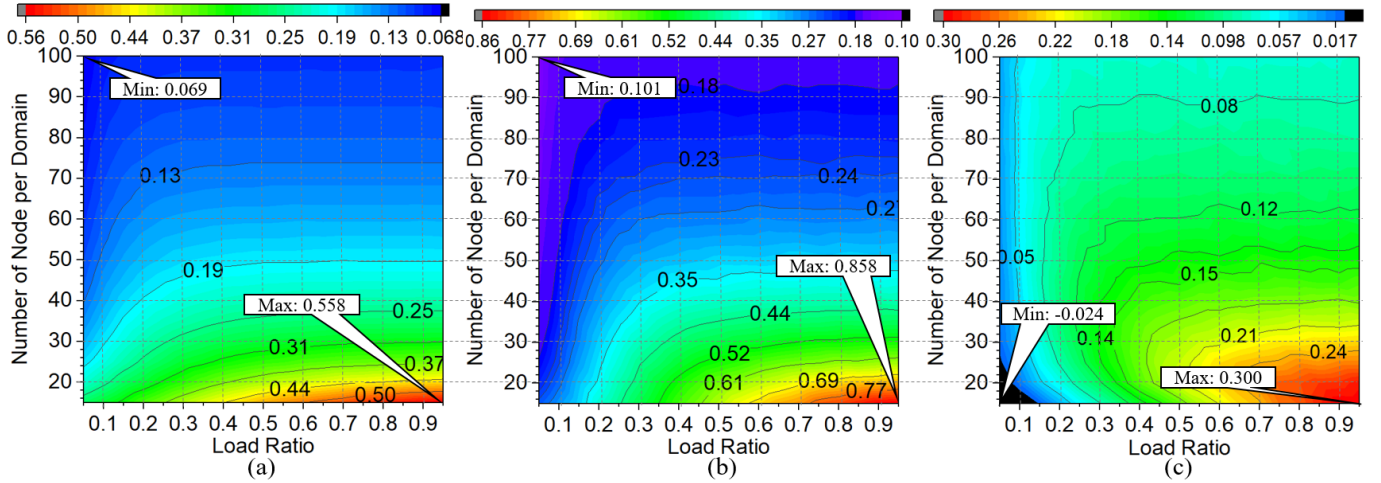

Fig. 7.   Evaluation of compression degree: (a) Node aggregation degree; (b) Link aggregation degree; (c) Degree difference between link and node.

C. GOAS Algorithm

We propose a benchmark algorithm named Global OAS (GOAS) algorithm to compare the performance with and without topology aggregation, as shown in Algorithm III. Different from OAS that observes the physical networks only through an aggregated view, GOAS is designed to have a global view, i.e., GOAS knows all the details of multi-domain physical networks. At Step 2, GOAS checks if the multi-domain physical networks could provide enough SF resources for the request $R(s, d, b, c)$. If not, this request is blocked at Step 13. If yes, each iteration in **FOR** loop (Steps 5-9) will find an anchor node satisfying specific SF requirement. In Step 7, if the segment is cross-domain, it is separated into two sub-segments by BNs and then RSA algorithm is executed on both sub-segments.

GOAS can be considered as a non-aggregated version of OAS. GOAS calculates in each segment from source to destination by using the shortest path and first-fit RSA algorithm. There are two important differences between GOAS and OAS: i) there is no information distortion in GOAS, because it directly considers physical networks. And ii) GOAS is able to consider the continuity constraint and the contiguity constraint [30] in EON directly. However, these constraints are introduced while calculating for the wheel matrix in topology aggregation, because the element of wheel matrix means the routing attributes on physical networks. Hence, OAS could only handle these constraints by wheel matrix indirectly.

D. Time-Complexity Analysis

In this subsection, we analyze the time complexity of benchmark GOAS, proposed OAS and K-OAS. The time cost should cover the period from the arrival of requests, to the end of resource allocation. For OAS and K-OAS, the time cost could be divided into several parts, i.e., the time cost of TA, the time cost of calculating the solution on aggregated topology, and the time cost to de-map the SFP to each single-domain. It's hard to evaluate the time cost of TA, because it depends on sync policy, the size of the single domain, the resources inside a single domain, the domain number of the

multi-domain networks, etc. To simplify, topology aggregation for OAS and K-OAS is skipped during complexity analysis. We assume that for a SFP request $R(s, b, d, c)$, $|T|$ is the average number of the node that satisfies the requirement of specific SF resource and $|L|$ is the number of links in multi-domain networks. $|F|$ is fixed number of frequency slots per link.

In GOAS, there are $|c| + 1$ segments requiring computation. In each segment, there are T options to T next optional anchors to be selected. In a single-source single-target problem, the time-complexity of shortest path and first-fit RSA algorithm is $O(|L| + |N|\log(|N|) + |F||E|)$ [31]. Then, Equation (19) shows the time complexity of GOAS.

$$T_{GSOAS} = O(|c||T|(|L| + |N|\log(|N|) + |F||E|)) \quad (19)$$

The time complexity of OAS can be derived from that of Dijkstra [32]. Compared to OAS, K-OAS maintains k optional uncompleted SFPs. Hence, Equations (20) and (21) show time complexity of OAS and K-OAS respectively. For simplicity, we do not consider the complexity of wheel matrix operations and aggregation procedures.

$$T_{OAS} = O(|c|(|E| + |V|\log(|V|))) \quad (20)$$
$$T_{K-OAS} = O(k|c|(|E| + |V|\log(|V|))) \quad (21)$$

VI. SIMULATION RESULTS

In this section, we present simulative results about the performance of OAS and K-OAS. We use Waxman model [33] to generate multi-domain topologies based on the Erdos-Renyi random graph model [34] by using BRITE [35]. Fig. 6 shows a 9-domain network where clustered nodes in the same color form a domain. Bigger colored circles are BNs of each domain.

The topology generation method of Waxman consists of two levels: i) node-level to generate a single domain and ii) domain-level to connect multiple domains. In the node-level, we assume that u and v are two random nodes in a single-domain topology, d_{uv} is the Euclidean distance between them, and L is the maximum distance between any two nodes. Equation (22) shows the probability to connect u
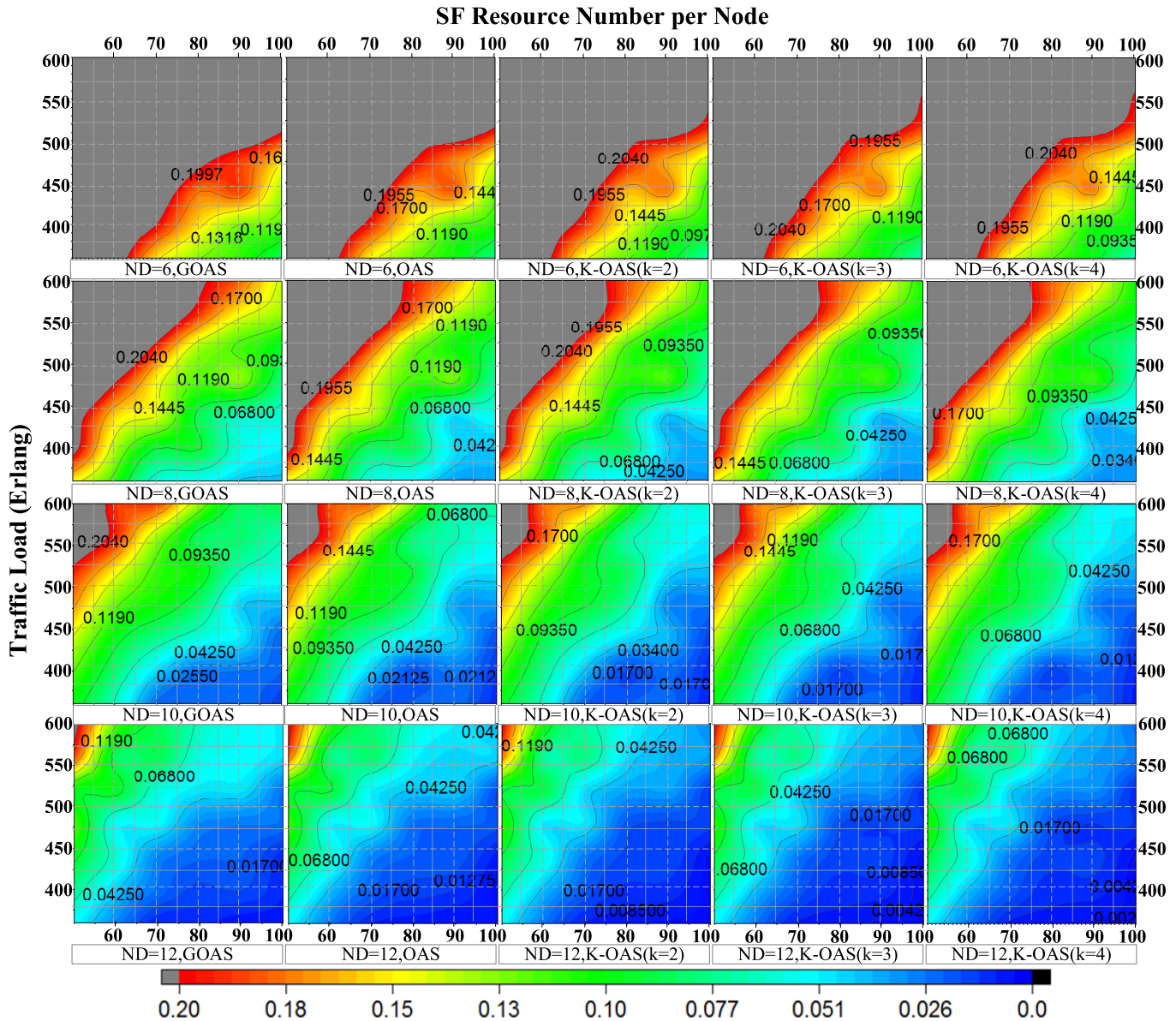
Fig. 8.   Blocking probability with different ND on different algorithms.

and v by a direct link where α and β are two coefficients. Domain-level generation is similar to node-level generation, where we considered each domain as a node, and set α and β as 0.15 and 0.2 to simulate inter-domain network connections.

$$P(u,v) = \alpha \cdot e^{\frac{-d_{uv}}{\beta \cdot L}} \tag{22}$$

We consider 384 frequency slots capacity for each intra-domain links. The domain operator usually sets the capacity of inter-domain links much larger than that of the internal links, which aims to guarantee the information exchange between domains. Therefore, we set capacity for inter-domain links as infinity to simplify the simulation. 6 types of SF $|S|$ and 10 types of SFC $|C|$ are defined. In a SFC $c = [(f, r_f, b_f)|f \in S]$, required SF types $|c|$ is a random integer between 1 and 3, r_f is a random integer between 1 and 4, and b_f is a random integer between 1 and 4. For a SFC request $R(s, d, b, c)$, source s and destination d are all BNs randomly selected from different domains. Bandwidth requirement b is the same as b_f in c. The

arrival and departure of requests follow Poisson process where μ is 0.5.

A. Aggregation Degree

Figure 7 shows node aggregation degree and link aggregation degree according to Equations (4-5) where the scope of load ratio is [0.05, 0.95] and the number of nodes per domain is [15, 100]. Load ratio means the probability for each node to carry SF resource. 547,670 domains under different settings are generated to evaluate information compression of SOTA. Figure 7 (a-b) show that aggregation degree of both node and link increases when the number of nodes per domain becomes smaller or the load ratio becomes larger. Explanation of this phenomenon is: number of nodes per domain affects denominator of Equation (4-5) and load ratio affects numerator of Equation (4-5). According to Fig. 7, the best node aggregation degree is 0.069, and the best link

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

YAN *et al.*: SERVICE FUNCTION PATH PROVISIONING WITH TOPOLOGY AGGREGATION IN MULTI-DOMAIN OPTICAL NETWORKS 11

aggregation degree is 0.101. Both best degrees reach minimum value when load ratio equals 0.05 and the number of nodes per domain equals 100. In short, in the best-case scenario, SOTA can compress 93.1% nodes and 89.99% links.

Figure 7(c) shows the difference of distribution between link aggregation degree and node aggregation degree, which aims to compare both degrees. Link aggregation degree is higher than node aggregation degree mostly, which proves that link information compression plays a main part in SOTA relatively. However, link aggregation degree is lower than node aggregation degree when load ratio is less than 0.15 and the number of nodes per domain is less than 25 (black bottom left corner of Fig. 7(c)), because extreme low load ratio will reduce the number of ASNs.

B. Blocking Probability

Figure 8 shows blocking of SFP provisioning with different Number of Domains (NDs), Number of SF resources per node, and traffic loads (for GOAS, OAS, and K-OAS algorithms). Color bar of Fig. 8 indicates the value of contour lines in the 4×5 subgraphs. For example, the gray color represents blocking larger than 20% and the dark blue color represents blocking less than 2.600%.

In each subgraph, blocking probability decreases when the number of SF resource per node increases, which means enough SF resources helps SFP allocation. Besides, larger traffic load would cause higher blocking probability obviously. In each column of 20 subgraphs vertically, blocking probability decreases when ND increases, which means more domains provide more SFP options. And in each row of 20 subgraphs horizontally, the performance of OAS is better than GOAS, and K-OAS performs better with increasing k. We can see the performance difference by the movement of contour lines and the emergence of dark blue color in the graphs. However, the difference is not very clear in the subgraphs of K-OAS($k = 3$) and K-OAS($k = 4$), which means larger k value doesn't always make obvious improvement but always costs more time. So there should be a trade-off when choosing the best k value.

SFP will occupy both SF resource and FS resource. The main reasons for SFP blocking could be summarized as two resource-lack limitations: limitation of FS on ASNs and limitation of SF on links. The explanation is as follows: all three algorithms (GOAS, OAS, and K-OAS) checks for enough SF resources to satisfy SFP requests. If satisfied, SFP request is handled by core logic of algorithms. Therefore, the performance difference among the algorithms comes from the differences in handling FS. Fig. 9 shows the relationship between SF Resource Number per Node (SRNN) and Ratio of blocked SFPs caused by SF limitation in all blocked SFPs (RSF). In Fig. 9, RSF becomes smaller when SRNN increases and smaller ND will cause larger RSF. When ND is 12 in each subgraph, RSF reaches the lowest value for all the algorithms (larger ND means more SF resources). In Fig. 9, RSF of OAS is slightly higher than GOAS. And RSF of K-OAS is higher slightly than OAS. These differences show that GOAS
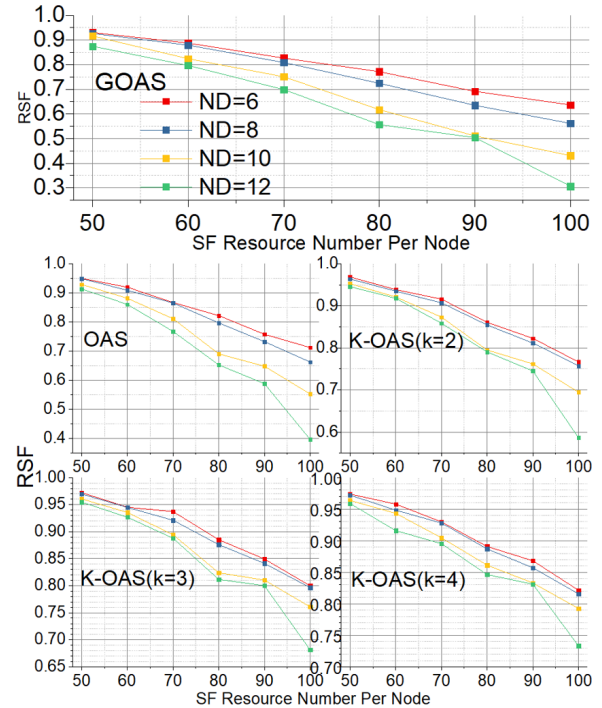

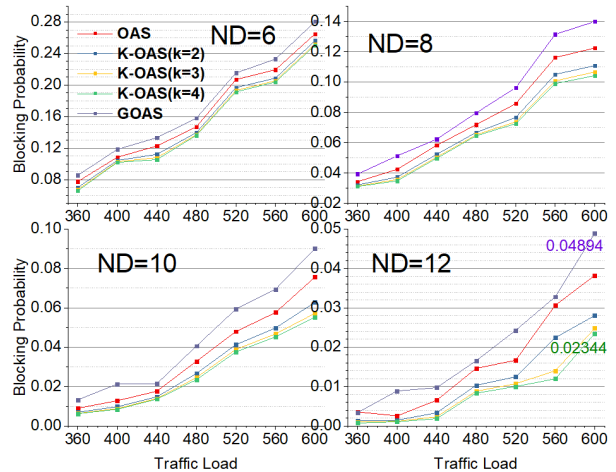Fig. 9.   Ratio of SF lack to blocked SFPs for different algorithms.

Fig. 10.   Blocking probability when SRNN equals 100.

has lowest and K-OAS ($k = 4$) has the highest SF resource utilization.

Figure 10 shows the differences in blocking ratio among the algorithms with SRNN is 100. This is a well-resourced scenario, and SFP request provisioning methods cause the reduction of blocking probability. In all 4 subgraphs of Fig. 10, K-OAS (k=4) has the lowest blocking probability compared to the other algorithms. With the increase of the ND, the gap among different algorithms becomes wider. In the scenario where ND equals 12, the ratio of blocking reduction between K-OAS (k=4) and GOAS reaches (0.04894-0.02344)/0.04894=52.10%.

C. Number of Crossed Domain

Cross-domain routing introduces higher calculation complexity with continuity constraint and contiguity constraint.
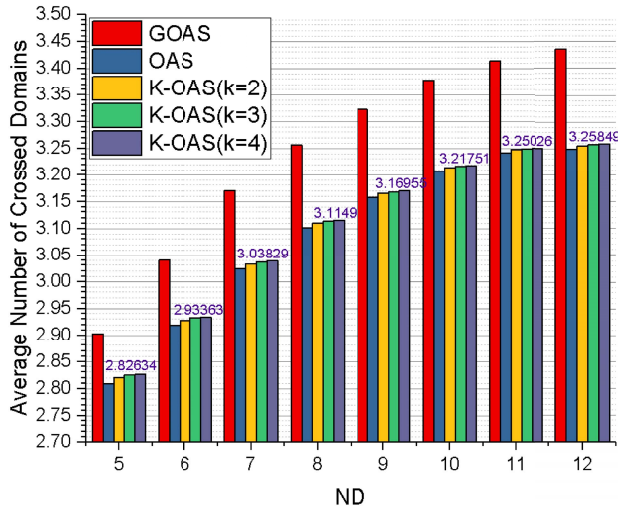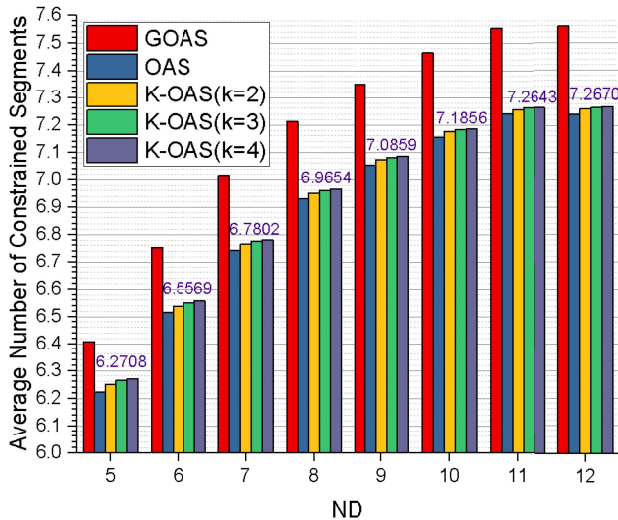
Fig. 11.   Average number of crossed domains.

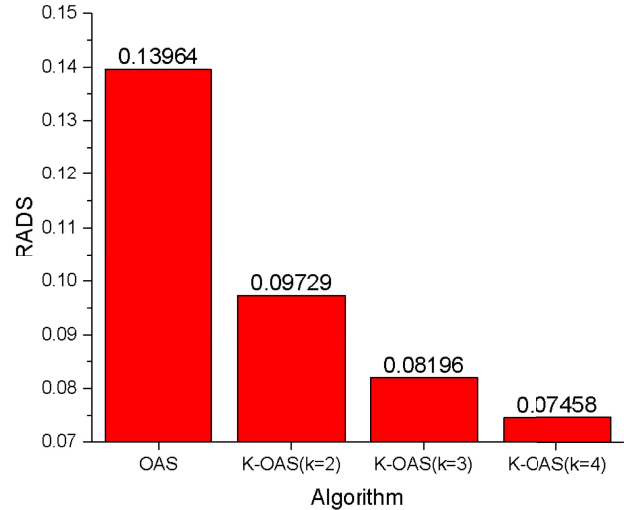Fig. 13.   Aggregation distortion evaluation for SOTA.

D. Constrained Segments

A SFP can be divided by anchor nodes into several segments, and one segment may cross multiple domains. This routing part that follows continuity constraint and contiguity constraint, named constrained segment, always starts or ends at anchor nodes or border nodes. Different constrained segments of a SFP in EON may use different frequency slots to carry SFP traffic, which is an important factor for building a SFP. Figure 12 shows the average number of constrained segments with the different number of domains where load ratio = 0.6. The number of the average constrained segments closely follow the average number of crossed domains. We observe that number of constrained segments of GOAS is higher than others and number for K-OAS slowly grows with the larger number of domains.

Fig. 12.   Average number of constrained segments of different algorithms.

E. Aggregation Distortion

After calculating SFP on aggregated topology, the allocation for SF resources and FS resources on physical networks may fail due to information distortion introduced by SOTA. For example, the multiple segments inside a domain may occupy the bandwidth of the same link. The available bandwidth on this link maybe not able to carry multiple occupations. However, this conflict cannot be found on aggregated topology. Fig. 13 shows the Ratio of blocked services caused by Aggregation Distortion to all blocked Services (RADS). We observe that RADS of OAS is higher than K-OAS and RADS of K-OAS becomes smaller as k increases (i.e., K-OAS tries k paths to allocate SFPs). Because OAS has less SFP options compared to K-OAS. In addition, the number of domains is positively correlated with RADS, i.e., more domains mean larger average number of constrained segments creating longer routing paths, which results in more distortion of information.

It also enhances the difficulty of cooperation among different domain operators due to billing and other aspects. Hence, reducing the average number of crossed domains for SFPs is an important aspect of the problem.

Figure 11 shows the number of crossed domains by GOAS, OAS, and K-OAS algorithms with varying NDs. We observe that the average number of crossed domains increases when the physical network has more domains. OAS and K-OAS benefit from L_A and L_{BA}, which leads to visit ASNs in the same domain. Besides, GOAS visits SNs according to distance and does not consider if it crosses domains. Hence, the average number of crossed domains of GOAS is higher than the other two.

OAS always finds the shortest path for resource allocation. If the shortest-path SFP is unavailable, then the request is blocked. On the contrary, K-OAS finds additional SFPs to reduce blocking probability. The number of crossed domains in the other *k-1* paths can larger than that of the shortest path. Hence, the average number of crossed domains of K-OAS is higher than that of OAS and increases with k.

VII. Conclusion

We propose SFC-Oriented Topology Aggregation (SOTA) algorithm to enable SF resource aggregation. We also propose the usage of a new data structure named *wheel matrix* to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

YAN *et al.*: SERVICE FUNCTION PATH PROVISIONING WITH TOPOLOGY AGGREGATION IN MULTI-DOMAIN OPTICAL NETWORKS 13

support calculation for resource allocation in an aggregated topology. Based on the *wheel matrix*, Ordered Anchor Selection (OAS) and k-paths OAS (K-OAS) algorithms are proposed to calculate cross-domain SFPs. And we also propose a basic algorithm without TA named Global OAS (GOAS) as the benchmark. Simulation results show that SOTA is able to aggregate topology containing SF resources and compress as much as 90% network information. OAS and K-OAS designed on aggregated topology are able to calculate SFPs efficiently when SF resource is enough. Compared to the benchmark, the proposed algorithms also show significant improvements in terms of the average number of crossed domains and ratio of aggregation distortion to blocked services. Besides, SF is a kind of node attribute, so SOTA, *wheel matrix*, and the proposed algorithms could also be applied to other similar scenarios where node attribute need to be abstracted.

REFERENCES

[1] C. Gao, H. C. Cankaya, and J. P. Jue, "Survivable inter-domain routing based on topology aggregation with intra-domain disjointness information in multi-domain optical networks," *J. Opt. Commun. Netw.*, vol. 6, no. 7, pp. 619–628, Jul. 2014.

[2] (May 19, 2014). *OTNv3 Amendment to E-NNI 2.0 OSPFv2-Based Routing, OIF-ENNI-OSPF-02.3*. [Online]. Available: http://www.oiforum.com/wp-content/uploads/OIF-ENNI-OSPF-02.3.pdf

[3] *Architecture for the Automatically Switched Optical Network*, Standard G.8080/Y1304, ITU-T, Jun. 2006. [Online]. Available: https://www.itu.int/rec/T-REC-G.8080

[4] G. Maier, C. Busca, and A. Pattavina, "Multi-domain routing techniques with topology aggregation in ASON networks," in *Proc. Int. Conf. Opt. Netw. Design Modeling*, Vilanova i la Geltru, Spain, Mar. 2008, pp. 1–6.

[5] (Dec. 2017). *Abstraction and Control of TE Networks (ACTN) Abstraction Methods, IETF draft-lee-teas-actn-abstraction-02*. [Online]. Available: https://datatracker.ietf.org/doc/draft-lee-teas-actn-abstraction/

[6] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with topology aggregation in delay-bandwidth sensitive networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 17–29, Feb. 2004.

[7] P. Wan, W. Jiao, and X. Wu, "A novel topology aggregation method for hierarchical routing in ASON network," in *Proc. IEEE GLOBECOM-IEEE Global Telecommun. Conf.*, Washington, DC, USA, Nov. 2007, pp. 2275–2279.

[8] Q. Wang, Y. Lu, and Y. Ji, "Topology aggregation method in WSON networks," in *Proc. 3rd IEEE Int. Conf. Broadband Netw. Multimedia Technol. (IC-BNMT)*, Beijing, China, Oct. 2010, pp. 345–348.

[9] R. Casellas *et al.*, "Dynamic virtual link mesh topology aggregation in multi-domain translucent WSON with hierarchical-PCE," in *Proc. 37th Eur. Conf. Expo. Opt. Commun.*, Geneva, Switzerland, 2011, pp. 1–3.

[10] M. Cai, L. Tang, and R. Wu, "A novel topology aggregation algorithm in multi-QoS restricted multi-domain optical networks," in *Proc. 7th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Wuhan, China, Sep. 2011, pp. 1–4.

[11] C. Gao, Y. Zhu, and J. P. Jue, "SRLG-aware topology aggregation for survivable multi-domain optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 11, pp. 1145–1156, Nov. 2013.

[12] C. Gao, M. M. Hasan, and J. P. Jue, "Domain-disjoint routing based on topology aggregation for survivable multidomain optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 12, pp. 1382–1390, Dec. 2013.

[13] C. Gao, H. C. Cankaya, and J. P. Jue, "Survivable inter-domain routing based on topology aggregation with intra-domain disjointness information in multi-domain optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 6, no. 7, pp. 619–628, Jul. 2014.

[14] A. A. Akyama, S. Sengupta, J. F. Labourdette, S. Chaudhuri, and S. French, "Reliability in single domain vs. multi domain optical mesh networks," in *Proc. IEEE/OSA Nat. Fiber Opt. Eng. Conf.*, Dallas, TX, USA, Oct. 2002, pp. 240–249.

[15] Y. Zhang, L. Guo, X. Wang, X. Zheng, and X. Wang, "Differentiated domain protection algorithm based on virtual topology graph in multi-domain optical networks," in *Proc. WRI Int. Conf. Commun. Mobile Comput.*, Yunnan, China, Jan. 2009, pp. 526–530.

[16] T. Li, H. Zhou, and H. Luo, "A new method for providing network services: Service function chain," *Opt. Switching Netw.*, vol. 26, pp. 60–68, Nov. 2017.

[17] X. Zheng and N. Hua, "Achieving inter-connection in multi-domain heterogeneous optical network: From PCE to SDON," in *Proc. Asia Commun. Photon. Conf.*, Beijing, China, 2013, pp. 1–3, Paper AW4I.2.

[18] (Oct. 2013). *Network Functions Virtualization (NFV); Use Cases, ETSI GS NFV 001 v1.1.1*. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf

[19] S. Rahman, T. Ahmed, M. Huynh, M. Tornatore, and B. Mukherjee, "Auto-scaling VNFs using machine learning to improve QoS and reduce cost," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[20] *Study on Architecture for Next Generation System*, document TR 23.799, release 14, version 7.0, 3GPP, 2015. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3008

[21] S. Rahman, A. Gupta, M. Tornatore, and B. Mukherjee, "Dynamic workload migration over optical backbone network to minimize data center electricity cost," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 570–579, Dec. 2017.

[22] *Problem Statement for Service Function Chaining*, doccument RFC7498, IETF, Nov. 2015. [Online]. Available: https://datatracker.ietf.org/doc/rfc7498/

[23] (Apr. 2018). *Service Function Path Establishment, IETF Draft-LAN-SFP-Establishment-05*. [Online]. Available: https://datatracker.ietf.org/doc/draft-lan-sfp-establishment/

[24] (Apr. 2018). *Multi-Domain Service Forwarding For NSH, IETF Draft-LI-SFC-NSH-Multi-Domain-04*, Apr. 2018. [Online]. Available: https://datatracker.ietf.org/doc/draft-li-sfc-nsh-multi-domain/

[25] B. Yan, Y. Zhao, X. Yu, W. Wang, and J. Zhang, "Service function-oriented topology aggregation in multi-domain inter-DC elastic optical networks," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, Mar. 2018, pp. 1–3, paper Th3F.3

[26] B. Sonkoly *et al.*, "Multi-domain service orchestration over networks and clouds: A unified approach," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 377–378, Sep. 2015.

[27] G. Sun, Y. Li, D. Liao, and V. Chang, "Service function chain orchestration across multiple domains: A full mesh aggregation approach," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 1175–1191, Sep. 2018.

[28] Q. Zhang, X. Wang, I. Kim, P. Palacharla, and T. Ikeuchi, "Vertex-centric computation of service function chains in multi-domain networks," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Seoul, South Korea, Jun. 2016, pp. 211–218.

[29] Q. Zhang, X. Wang, P. Palacharla, and T. Ikeuchi, "Distributed service orchestration: Interweaving switch and service functions across domains," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Palo Alto, CA, USA, Nov. 2016, pp. 13–18.

[30] V. Lopez and L. Velasco, "Routing and spectrum allocation," in *Elastic Optical Networks: Architectures, Technologies, and Control.* Cham, Switzerland: Springer, 2016, ch. 4.

[31] B. Yan *et al.*, "Tidal-traffic-aware routing and spectrum allocation in elastic optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 11, pp. 832–842, Nov. 2018.

[32] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM (JACM)*, vol. 34, no. 3, pp. 596–615, Jul. 1987.

[33] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[34] P. Erdos and A. Rényi, "On the evolution of random graphs," *Pub. Math. Inst. Hungarian Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.

[35] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proc. MASCOTS, Proc. 9th Int. Symp. Modeling, Anal. Simulation Comput. Telecommun. Syst.*, Cincinnati, OH, USA, Aug. 2001, pp. 346–353.