Data-Oriented Task Scheduling in Fixed- and Flexible-Grid Multilayer Inter-DC Optical Networks: A Comparison Study

Ping Lu, *Student Member, IEEE*, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—It is known that in multilayer inter-datacenter optical networks (ML-IDCONs), network services can generate numerous data-oriented tasks (DoTs) that require not only inter-DC data transfers but also data processing in the DCs. In this paper, we perform a comparison study on the scheduling of DoTs (*i.e.*, setting up lightpaths on fiber links for inter-DC data transfers and scheduling DoT buffering/processing in DCs) in fixed- and flexible-grid ML-IDCONs. We propose a DoT scheduling algorithm that can work well for both types of networks. Specifically, for each DoT, the algorithm first tries to serve it using the residual bandwidth in the IP layer. This is achieved by expanding the time-expanded network (TEN) approach and transforming the store-and-forward (SnF) assisted DoT scheduling problem into a minimum-cost maximum-flow (MCMF) problem. Then, by solving the MCMF problem, we find the way to maximize the data transfer throughput and minimize the total DC storage usage simultaneously. Next, if the obtained data transfer throughput is not sufficient for the DoT, the algorithm tries to build lightpath segments for it based on the branch and bound scenario. Extensive simulations are conducted to evaluate the proposed algorithm's performance in fixed- and flexible-grid ML-IDCONs, and also compare it with three benchmarks. Simulation results indicate that for DoT scheduling, the flexible-grid ML-IDCON can outperform fixed-grid ones in terms of the blocking probability, energy consumption of transponders, and usage of DC storage, and our algorithm achieves lower blocking probability than the benchmarks with comparable or higher time-efficiency.**

***Index Terms*—Data-oriented task scheduling, Inter-datacenter optical networks, Multilayer networking, Elastic optical networks (EONs), Store-and-forward scheme.**

I. Introduction

NOWADAYS, due to the fast development of cloud computing and Big Data applications, there has been an ever accelerating expansion of datacenter (DC) networks globally [1]. Large enterprises, such as Google and Facebook, are adopting geographically distributed (geo-distributed) DCs to ensure high-quality and reliable services to their customers world-wide [2, 3]. In a geo-distributed DC system, applications such as data analytics, publish-subscribe (Pub-Sub) services, and inter-DC video transfer, may invoke numerous data-oriented tasks (DoTs) that not only require to transfer bulk data over the network interconnecting the DCs but also need to get data processed in the DCs. Recently, with the increase of these DoTs, inter-DC traffic has become more and more dominant in geo-distributed DC systems [2]. To carry such inter-DC traffic cost-effectively, optical networks are the only feasible physical infrastructure [4], and the multilayer inter-DC optical network (ML-IDCON) architecture that is based on IP-over-optical has been frequently used [5, 6].

In general, the procedure of scheduling DoTs in an ML-IDCON consists of two operations: 1) setting up lightpaths for inter-DC data transfers, and 2) reserving IT resources (*e.g.*, CPU cycles and memory) on the DCs for data buffering/processing. Note that, DoTs are usually delay-tolerant and malleable [7, 8], which means that their service provisioning has several unique characteristics to make it intrinsically different from that of flow-oriented bandwidth-fixed/variable lightpaths [8]. Firstly, the data transfer of a DoT does not have to be continuous or based on one end-to-end lightpath. Specifically, we can establish several lightpath segments to accomplish the data transfer in a store-and-forward (SnF) manner [9, 10]. In other words, the IP routers at intermediate DCs can realize optical-to-electrical-to-optical (O/E/O) conversions and leverage the storage space of the DCs to buffer the DoT's data for future transfer opportunities, when the corresponding output links are busy. Secondly, when serving a DoT, we need to allocate not only the spectrum resources in fiber links but also the storage space (*i.e.*, for SnF) and data processing resources in DCs, while to provision a flow-oriented lightpath, we only need to consider the spectrum resources.

Fig. 1 shows the feasible DoT scheduling schemes in an ML-IDCON, which consists two fiber links and three DC nodes. Each DC node consists of a DC, an IP router and a wavelength-selective switch (WSS). Hence, if we want to provision a DoT from *DC* 1 to *DC* 3, there are three feasible scheduling schemes, *i.e.*, a direct transparent lightpath from *DC* 1 to *DC* 3, a direct translucent lightpath that encounters an O/E/O conversion at *DC* 2, and an indirect SnF-assisted scheme that transfers the data from *DC* 1 to *DC* 2, buffers it there for a while, and then transfers it to *DC* 3. Therefore, we can see that the scheduling of a DoT is much more flexible and thus complex than the service provisioning of a flow-oriented bandwidth-fixed/variable lightpath. This is because we need to determine the scheme of lightpath transmission or DC buffering for the DoT in each time slot (TS) according to instantaneous network status and make sure that it can reach its destination DC before the deadline. Intuitively, for scheduling DoTs in an ML-IDCON, we need to solve the problem of time-dependent lightpath segmentation and establishment by leveraging the three feasible schemes in Fig. 1.

P. Lu and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

Manuscript received January 16, 2017.

Previously, a few studies have considered the service provisioning of flow-oriented bandwidth-fixed/variable lightpaths in [7, 11–16]. Although both immediate reservation (IR) and advance reservation (AR) schemes have been addressed in these studies, their network models are intrinsically different from ours in this paper, since they all assumed that the data transfer on a lightpath has to be continuous until all the data has been transmitted (*i.e.*, the lightpath's hold-on time expires). On the other hand, the scheduling schemes of DoTs in fixed- and flexible-grid optical networks have been investigated in [8, 17, 18]. However, none of them has considered the indirect SnF scheme in DoTs scheduling. Note that, according to [9], the indirect SnF scheme can significantly improve the bandwidth utilization in an inter-DC network and make the scheduling of DoTs more adaptive and flexible.

Recently, Feng *et al.* [19] tried to improve the data transfer throughput in fixed-grid wavelength-division multiplexing (WDM) networks with the SnF scheme. Nevertheless, their proposal still bears a few drawbacks. First of all, as we will show later in this work, both the effectiveness and time-efficiency of the proposed algorithm in [19] can be further optimized. Secondly, the work in [19] only focused on the optical layer but did not consider the multilayer network architecture. As a result, it did not try to minimize the costs in IP and optical layers when scheduling the DoTs. Note that, the SnF scheme would require extra transponders in the IP routers and thus increase energy consumption, and moreover, it consume more storage space in the DCs too. Hence, the advantage of the SnF scheme cannot be justified without taking these extra costs into consideration. Finally, the work in [19] did not consider the flexible-grid elastic optical networks (EONs) [20–22]. It is known that since EONs utilize bandwidth-variable transponders (BV-Ts) and WSS' (BV-WSS') to allocate optical spectra with a granularity of 12.5 GHz or even smaller, they make the optical layer more adaptive and thus are more suitable for ML-IDCONs [4], especially when the inter-DC traffic is highly dynamic. To the best of our knowledge, there has been no previous work that covers both fixed- and flexible-grid ML-IDCONs comprehensively and develops an effective and time-efficient algorithm to minimize the usages of transponders and DC storage space when scheduling the DoTs in ML-IDCONs.

In this paper, we conduct a comparison study on the scheduling of DoTs (*i.e.*, setting up lightpaths on fiber links for inter-DC data transfers and scheduling DoT buffering/processing in DCs) in fixed- and flexible-grid ML-IDCONs. We carefully consider the characteristics of these two types of ML-IDCONs, and propose a DoT scheduling algorithm that can work well for both of them. The algorithm handles the DoTs sequentially according to their delay constraints (*i.e.*, deadlines). Specifically, for each DoT, the algorithm first tries to serve it using the residual bandwidth in the IP layer. To achieve this, we expand the time-expanded network (TEN) approach developed in [9], and transform the SnF-assisted DoT scheduling problem into a minimum-cost maximum-flow (MCMF) problem. Then, by solving the MCMF problem, we find a way that can maximize the data transfer throughput and minimize the total DC storage usage simultaneously. Next, if the obtained data transfer throughput is not sufficient for the DoT, the algorithm tries to build lightpath segments for it by leveraging the branch and bound scenario, *i.e.*, searching for a lightpath segmentation scheme with the smallest cost due to extra usages of transponders and DC storage space on K shortest path candidates.

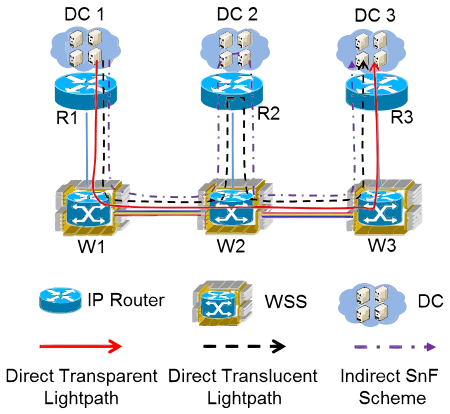


Fig. 1. Feasible DoT scheduling schemes in an ML-IDCON.

The rest of the paper is organized as follows. Section II provides a brief survey of the related work. We describe the problem of DoT scheduling in ML-IDCONs in Section III, and our proposed DoT scheduling algorithm is discussed in Section IV. Then, Section V presents the performance evaluation with simulations. Finally, we summarize the paper in Section VI.

II. Related Work

Previously, using inter-DC packet networks as the background, various studies [9, 23–27] have considered the DoT scheduling problem. Joe-Wong *et al.* [23] proposed a distributed algorithm to schedule DoTs, with the objective of minimizing the costs of energy consumption and bandwidth utilization. The authors of [24] designed a joint data/task placement algorithm to reduce the latency of inter-DC data analytics. The work in [9] suggested that the data transfer throughput of DoTs can be improved with the SnF scheme. Then, the TEN approach, which transforms a time-varying network into a static one by expanding the network's topology along the time axis, was adopted to analyze the performance of SnF-assisted DoT scheduling in [25]. In [26, 27], we leveraged the Lyapunov optimization technique to design online and distributed algorithms to schedule DoTs with anycast and multicast requirements, respectively. However, since the resource allocation constraints in the optical layer (*e.g.*, the wavelength non-overlapping and continuity constraints [20, 28]) have not been considered in these studies, the algorithms developed in them cannot be applied to ML-IDCONs.

For optical networks, several data transfer schemes have been studied in [7, 8, 13, 15, 17, 18, 29, 30]. In [29], the authors investigated how to schedule the deadline-driven data transfers in fixed-grid WDM networks to improve wavelength utilization. Lu *et al.* [13] studied how to realize spectral-efficient AR with intelligent request scheduling in EONs,

and the authors extended their work to consider AR and IR requests jointly in [15]. However, a DoT is different from the flow-oriented IR/AR lightpath request in a few aspects. Firstly, to serve a DoT, we need to allocate multiple types of network resources (*i.e.*, optical spectra, DC storage space, and data processing resources) in a correlated and time-varying way according to instantaneous network status. Secondly, the DoT can be served by leveraging the SnF scheme, which would not be beneficial to a flow-oriented IR/AR lightpath request since its data transfer has to be continuous. Moreover, we hope to point out that the problem of DoT provisioning with SnF cannot be solved by simply extending the approaches developed to address the classical translucent lightpath provisioning with O/E/O conversions [31]. This is because for DoT provisioning with SnF, we have to optimize the scheduling of DoTs in the time domain in an end-to-end manner, which is not feasible by simply extending the classical O/E/O conversion studies.

The problem of bandwidth-variable AR in WDM networks was considered in [7], where several algorithms were proposed to minimize the overall data transfer time. With the mutual backup model, the work in [17] designed several algorithms to minimize the time required for finishing a regular DC backup in fixed-grid ML-IDCONs. The studies in [8, 18] investigated how to schedule DoTs in EONs to recycle the spectrum fragments generated by flow-oriented lightpaths. Yi *et al.* [30] considered the budge-optimized scheduling of DoTs with fixed bandwidth requirements in fixed-grid ML-IDCONs. Nevertheless, none of these studies tried to use the SnF scheme to schedule DoTs. Although the work in [19] tried to improve the data transfer throughput in fixed-grid WDM networks with the SnF scheme, the proposed algorithm still bears a few drawbacks. Firstly, the algorithm finds a feasible lightpath segmentation scheme by traversing all the possible schemes on K shortest path candidates. This, however, makes its time complexity as $O(K \cdot 2^n)$, where n is the largest hop-count of a path candidate. Hence, the algorithm cannot be solved in polynomial time and thus has relatively high complexity. Secondly, the algorithm did not try to balance the resource usages in the inter-DC network, which might increase the blocking probability of service provisioning. Lastly, but not least, the work did not cover EONs in its network model.

The performance comparisons of fixed- and flexible-grid optical networks have been discussed in [32] for the spectral efficiency of lightpaths, and in [33] for the energy consumption and capital cost. However, none of these investigations have considered the scheduling of DoTs in ML-IDCONs.

III. DoT Scheduling in ML-IDCONs

A. Network Model

We consider a discrete-time system for the ML-IDCON, which means that the time axis is divided into TS' with a fixed duration (*i.e.*, Δt seconds) and all the network operations are performed at the beginning of a TS. As a network element in the optical layer (*e.g.*, a BV-WSS) can take hundreds of milliseconds in each reconfiguration [34], we assume that Δt is much longer than such a reconfiguration transition. The DoT scheduling considers a total period of T TS', *i.e.*, we have each TS $t \in [1, T]$. An IP-over-optical network architecture is considered for the ML-IDCON, which is denoted as $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, where $\mathcal{D}$ and $\mathcal{E}$ represent the sets of DCs and inter-DC fiber links, respectively. And each DC connects to a WSS through an IP router. The corresponding IP layer is denoted as $\mathcal{G}_e = (\mathcal{D}, \mathcal{E}_e)$, where $\mathcal{E}_e$ includes the virtual links realized with established lightpaths in the ML-IDCON.

We assume that the total available spectrum on a fiber link is W GHz, which is a constant for both fixed- and flexible-grid ML-IDCONs. Without loss of generality, the spectrum can be divided into N wavelength channels or frequency slots (FS') in fixed- or flexible-grid ML-IDCONs, respectively, while each channel or FS occupies Δw GHz bandwidth and can provide a capacity of Δb Gb/s when certain modulation format is used [35]. Here, N, Δw and Δb can take different values to adapt to various networking scenarios. For instance, the value of N in the flexible-grid ML-IDCON would be larger than that in the fixed-grid ML-IDCON, since the channel width is narrower. Also, when the optical signal's modulation-level is higher, we can obtain a higher Δb with the same Δw, but at the same time, the transmission reach of the optical signal will decrease [36]. In this work, we follow the model presented in [36] to determine the transmission reach.

To represent instantaneous network status, we introduce a binary variable $w_{n,e}(t)$ to denote the availability of a wavelength channel or an FS, *i.e.*, we have $w_{n,e}(t) = 1$ if the n-th wavelength channel or FS on link $e \in \mathcal{E}$ is available at TS t, and $w_{n,e}(t) = 0$ otherwise. The number of available transponders in DC node i at time t is denoted as $M_i(t)$. In the fixed-grid ML-IDCON, a transponder covers a single wavelength channel, while a BV-T in the flexible-grid ML-IDCON can cover up to F FS'. Moreover, since the channel width Δw in the flexible-grid ML-IDCON is relatively small, we reserve a guard-band of f_g FS' for each lightpath, while for the fixed-grid ML-IDCON, there is no need to introduce such a guard-band [32]. Note that, we do not consider the nonlinear effects in the flexible-grid ML-IDCON here for simplicity, and assume that nonlinear crosstalk can be compensated by the guard-bands. Meanwhile, we notice that the network model in [37], which considers the nonlinear effects, might allow more efficient network modeling and optimization, and we will try to incorporate it in our future work.

As for the IP layer $\mathcal{G}_e$, we denote the available bandwidth on a virtual link $(u, v) \in \mathcal{E}_e$ as $b_{u,v}(t)$, which is initialized as 0 when there is no lightpath to connect DC nodes u and v. The computing resources in a DC can be quantified in different ways, *e.g.*, in terms of virtual machines and CPU cores. Since we focus on DoT scheduling rather than the actual task processing techniques, the available computing resources in *DC* i at TS t are normalized and denoted as $C_i(t)$ in units [38]. Meanwhile, the available storage space in *DC* i at TS t is denoted as $S_i(t)$ in Gigabytes (GB).

B. Service Model

We assume that there are J DoTs that need to be served in the ML-IDCON within $t \in [1, T]$. Each DoT contains a large amount of data that is generated in its source DC and

should be transferred to and processed in its destination DC. We use a tuple $\langle s_j, d_j, A_j, \delta_j, t_j^s, t_j^e, D_j \rangle$ to represent the j-th DoT, where s_j and d_j are the source and destination DCs, respectively, A_j is the amount of data in GB to be transferred and processed, δ_j is the required computing resources to process 1 GB data of the DoT, t_j^s and t_j^e are the start and end time of the DoT's scheduling window, respectively, and D_j is the maximum sojourn time, which equals to $t_j^e - t_j^s + 1$. Note that, the DoT should be transferred and processed before the deadline t_j^e, and considered as blocked otherwise.

Meanwhile, we consider a practical scenario in which there might be flow-oriented lightpaths in the ML-IDCON too. Since the data transfer of the flow-oriented lightpaths should be continuous, we assume that their priority is higher than the DoTs and should be considered as the background traffic in DoT scheduling [8]. Specifically, the flow-oriented lightpaths are randomly generated and then served before the DoT scheduling starts, and we can obtain the spectrum utilization of the ML-IDCON before the DoT scheduling as η, *i.e.*,

$$\eta = 1 - \frac{1}{T \cdot N \cdot |\mathcal{E}|} \sum_{n,e,t} w_{n,e}(t). \tag{1}$$

In the DoT scheduling, we should schedule the data transfer by either utilizing the residual bandwidth in the IP layer or establishing new lightpaths in the optical layer, and then perform task processing in the destination DCs. Since a DoT can be blocked if it cannot be transferred and processed by its deadline, we should try to balance the resource usages in the ML-IDCON to minimize the blocking probability.

IV. Resource-aware DoT Scheduling Algorithm

A. Overall Procedure

Algorithm 1 shows the overall procedure of our resource-aware DoT scheduling algorithm. For initialization, *Lines* 1-4 calculate K shortest paths (*i.e.*, in transmission distance) for each DC pair (i, j) in $\mathcal{G}$, and store the paths in set $\mathcal{P}_{i,j}$. Then, for each TS t, *Line* 6 sorts all the new DoTs (*i.e.*, those with $t_j^s = t$) in ascending order of their deadlines. The for-loop that covers *Lines* 7-15 schedules each DoT in the sorted order. More specifically, in each loop, *Line* 8 first tries to serve the DoT using the residual bandwidth in the IP layer $\mathcal{G}_e$ (*i.e.*, by applying *Algorithm* 2 to schedule the DoT). If this cannot be done, we apply *Algorithm* 3 in *Line* 10 to build SnF-assisted lightpath segments for serving the DoT. Finally, if the DoT still cannot be served, we will block it. In the following subsections, we first design *Algorithms* 2 and 3 for the flexible-grid ML-IDCON, and then we show that with only minor modifications, the proposed algorithms also work well for the fixed-grid ML-IDCON.

B. TEN-based DoT Scheduling in the IP Layer

As discussed above, for each DoT, we first try to transfer its data using the residual bandwidth in the IP layer. By leveraging the storage resources in intermediate DCs, we can realize the data transfer with the SnF scheme. Here, when designing the SnF scheme, we should consider two important factors. First of all, we should try to get all the data of the DoT transferred and processed successfully. Secondly, as the storage space can be limited in each DC, we should try to reduce the storage usage of the DoT over time and save certain space for subsequent DoTs. Hence, the DoTs' blocking probability can be reduced. These two factors can be taken care of by leveraging the TEN approach [9], which transforms a time-varying network into a static one by expanding it along the time axis.

We expand the TEN approach in [9] to optimize the DoT's usages on residual IP bandwidth, and storage and computing resources, and transform the DoT scheduling into a minimum-cost maximum-flow (MCMF) problem. Note that, since multi-path data transfer can lead to packet disorder at the destination DC, we do not consider it in this work and assume at any given TS, the data transfer of each DoT only use one lightpath. For the j-th sorted DoT provided by *Algorithm* 1, we first compute K shortest paths $\mathcal{P}^e_{s_j,d_j}$ from s_j to d_j in the IP layer $\mathcal{G}_e$. If we denote the hop-count of a path $p \in \mathcal{P}^e_{s_j,d_j}$ as $|p|$ (*i.e.*, there are $|p|$ virtual links or lightpaths on p), there are $|p|+1$ DC nodes along the path. The nodes can be indexed as $1, 2, \cdots, |p|+1$, and we denote the DC in the u-th node along p as $d_p(u)$. Then, we try to find a path in $\mathcal{P}^e_{s_j,d_j}$ such that the DoT's data can be transferred to and processed in its destination DC before the deadline. Specifically, we first build a TEN-based virtual topology based on $\mathcal{P}^e_{s_j,d_j}$ with a 4-Step scenario:

- **Step 1**: We replicate a path $p \in \mathcal{P}^e_{s_j,d_j}$ for D_j times. Each replica, denoted as $\mathcal{G}^t_{e,p}$, represents the network status in a specific TS $t \in [t_j^s, t_j^e]$, and the i-th node in replica $\mathcal{G}^t_{e,p}$ is referred to as v_i^t. The available bandwidth on virtual link $(d_p(i), d_p(i+1))$ in TS t is denoted as $B^p_{(v_i^t, v_{i+1}^t)}$, which indicates the amount of data that can be transferred on path p in TS t, *i.e.*,

$$B^p_{(v_i^t, v_{i+1}^t)} = \frac{b_{d_p(i),d_p(i+1)}(t) \cdot \Delta t}{8}, \forall i \in [1, |p|], t \in [t_j^s, t_j^e]. \tag{2}$$

- **Step 2**: We expand each $\mathcal{G}^t_{e,p}$ by adding an extra node V_t and a direct link from $v^t_{|p|+1}$ to V_t in it. The link is introduced to transform the computing resource constraint in destination DC d_j into a bandwidth constraint. Then, the bandwidth on the newly-added link is defined as the amount of data that can be processed in d_j in TS t, *i.e.*,

$$B^p_{(v^t_{|p|+1}, V_t)} = \frac{C_{d_j}(t)}{\delta_j}, \quad \forall t \in [t_j^s, t_j^e]. \tag{3}$$

- **Step 3**: We add links to connect adjacent replicas, *i.e.*, using a direct link (v_i^t, v_i^{t+1}) to connect v_i^t and v_i^{t+1}, and transform the storage space constraint into a bandwidth constraint. Specifically, we define

$$B^p_{(v_i^t, v_i^{t+1})} = S_{d_p(i)}(t), \quad \forall i \in [1, |p+1|], t \in [t_j^s, t_j^e - 1]. \tag{4}$$

- **Step 4**: We add a source node S and connect it to v_1^1 with a direct link, whose bandwidth is set as

$$B^p_{(S, v_1^1)} = A_j. \tag{5}$$

The TEN-based virtual topology can be denoted as $\hat{\mathcal{G}}_e = (\hat{\mathcal{D}}_e, \hat{\mathcal{E}}_e)$, which consists of $[(|p|+2) \cdot D_j + 1]$ nodes and $[(2 \cdot |p| + 3) \cdot D_j - |p| - 1]$ links. To balance the DoT's storage usage in the DCs over time, we introduce c_e to denote the unit

Algorithm 1: Resource-aware DoT Scheduling

1 **for** *each DC pair* (i, j) *in* $\mathcal{G}$ **do**
2 pre-compute K shortest paths;
3 store the paths in $\mathcal{P}_{i,j}$;
4 **end**
5 **for** $t = 1$ *to* T **do**
6 sort all the new DoTs in ascending order of their deadlines $\{t_j^e\}$;
7 **for** *each sorted DoT (*w.l.o.g., *the* j*-th one)* **do**
8 apply *Algorithm* 2 to serve the DoT using TEN-based approach in IP layer $\mathcal{G}_e$;
9 **if** *the DoT cannot be served* **then**
10 apply *Algorithm* 3 to build SnF-assisted lightpath segments for serving the DoT;
11 **if** *the DoT still cannot be served* **then**
12 block the DoT;
13 **end**
14 **end**
15 **end**
16 **end**

storage cost of buffering data in a DC, *i.e.*, transferring data on a link $e = (v_i^t, v_i^{t+1})$ in $\hat{\mathcal{G}}_v$. Here, c_e is defined as follows.

$$c_e = \begin{cases} \frac{1}{S_{d_{p(i)}}(t)}, & e \in \{(v_i^t, v_i^{t+1}) : i \in [1, |p|+1], t \in [t_j^s, t_j^e - 1]\}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Hence, when we perform MCMF-based routing and bandwidth allocation, DCs with more storage space will be preferred to be selected as intermediate DCs for the SnF scheme, which helps to balance the storage space usages of the DCs.

Fig. 2 gives an intuitive example to explain the aforementioned procedure. Here, we first generate the TEN-based virtual topology for a path that consists of 3 DCs and 2 links over 3 TS', and then find the MCMF in the virtual topology to solve the DoT scheduling. Each TS is assumed to be 5 minutes, the DoT has 1000 GB data to transfer (*i.e.*, $A_j = 1000$ GB), and all the DCs have 1000 GB storage and 500 units of computing resources over the three TS'. Fig. 2(a) shows the residual bandwidth on the links in each TS, and the TEN-based virtual topology obtained by the 4-Step scenario is illustrated in Fig. 2(b), where the two-element tuple aside each link marks its available bandwidth and unit storage cost, respectively. For instance, the tuple $\langle 1500, 0\rangle$ aside link (v_1^1, v_2^1) indicates that we can transfer 1500 GB data on this link without any storage cost. With the virtual topology in Fig. 2(b), if we want to transfer the maximum amount of data over the three TS' with the minimum storage cost, we just need to calculate the MCMF in it for $S \to V_3$. The MCMF is in Fig. 2(c), where the number aside each link is the amount of data that should be transferred through it. Based on the MCMF, we can schedule the data processing/buffering for the DoT. Specifically, we should deliver 1000 GB data from *DC* 1 to *DC* 2 and store it there in TS $t = 1$, transfer 750 and 250 GB data from *DC* 2 to *DC* 3 in TS' $t = 2$ and $t = 3$, respectively, and let *DC* 3 process 500 GB data in TS' $t = 2$ and $t = 3$.

Algorithm 2 shows the pseudo-codes of the procedure mentioned above, *i.e.*, to schedule the j-th DoT in the IP layer with the SnF scheme. *Lines* 3-4 generate a TEN-based

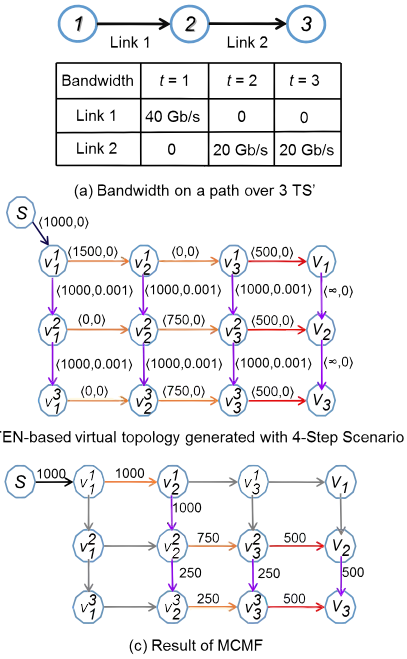


Fig. 2. Example on transforming DoT scheduling into MCMF.

virtual topology based on path $p \in \mathcal{P}^e_{s_j,d_j}$ and find the MCMF in it with the polynomial-time algorithm developed in [39]. Here, we use $\{b^p_{(u,v)}\}$ to denote the MCMF on path p, *i.e.*, the amount of data that should be transferred on link (u, v) in $\hat{\mathcal{G}}_e$. Note that, $b^p_{(S,v_1^1)}$ is actually the total amount of data that should be transferred in the IP layer of ML-IDCON within $[t_j^s, t_j^e]$. Hence, if we have $b^p_{(S,v_1^1)} \geq S_j$, the DoT can be scheduled with the residual bandwidth in the IP layer. Then, as shown in *Lines* 5-13, we perform DoT scheduling in the ML-IDCON based on $\{b_{(u,v)}\}$ in each TS t, *i.e.*, transferring, storing and processing the data of j-th DoT accordingly.

Since the time complexity of the algorithm in [39] to find the MCMF is $O(|E| \cdot \log(|V|) \cdot (|E| + |V| \cdot \log(|V|)))$, where V and E are the sets of nodes and links in the topology, respectively, the time complexity of *Algorithm* 2 is $O(|\mathcal{D}|^2 \cdot D_j^2 \cdot (\log(|\mathcal{D}| \cdot D_j))^2)$.

C. SnF-assisted Lightpath Segmentation for DoT Scheduling in the Optical Layer

If *Algorithm* 2 cannot serve the DoT, we will try to provision it by setting up new lightpaths in the optical layer. Note that, when the ML-IDCON is highly loaded, an end-to-end transparent lightpath might not always be feasible if we want to transfer the data from the DoT's source DC to its destination DC directly. Therefore, we design a SnF-assisted lightpath segmentation algorithm to perform DoT scheduling in the optical layer. Specifically, we first try to build several lightpath segments and then use them to realize the data transfer with the SnF scheme, *i.e.*, buffering the data on certain intermediate DCs and establishing time-dependent lightpath segments accordingly. However, the SnF-assisted lightpath segmentation would use more transponders and storage space in the IP routers and DCs, respectively. Considering the fact

Algorithm 2: TEN-based DoT Scheduling in IP Layer

1 compute K shortest paths for $s_j \to d_j$ in $\mathcal{G}_e$ and store them in $\mathcal{P}^e_{s_j,d_j}$;
2 **for** *each* $p \in \mathcal{P}^e_{s_j,d_j}$ **do**
3 generate virtual topology $\hat{\mathcal{G}}_e$ with 4-Step scenario;
4 get the MCMF $\{b^p_{(u,v)}\}$ for flow $S \to V_{t^e_j}$ with the algorithm in [39];
5 **if** $b^p_{(S,v^1_1)} \geq A_j$ **then**
6 **for** *each TS* $t \in [t^s_j, t^e_j]$ **do**
7 transfer $b^p_{(v^t_i, v^t_{i+1})}$ data of j-th DoT on link $(d_p(i), d_p(i+1))$ in TS t;
8 store $b^p_{(v^t_i, v^{t+1}_i)}$ data of j-th DoT in DC i in TS t;
9 process $b^p_{(v^t_{|p|+1}, V_t)}$ data of j-th DoT in destination DC in TS t;
10 update $\{b_{u,v}(t)\}, \{S_i(t)\}$ and $\{C_i(t)\}$;
11 **end**
12 **break**;
13 **end**
14 **end**

Algorithm 3: SnF-assisted Lightpath Segmentation for DoT Scheduling in Optical Layer

1 **for** *each* $p \in \mathcal{P}_{s_j,d_j}$ **do**
2 $X_u = \emptyset,\ Y_u = \emptyset,\ \forall u \in [1, |p|+1]$;
3 $\hat{t}^s_u = t^s_j,\ \hat{t}^e_u = t^s_j,\ \forall u \in [1, |p|+1]$;
4 $\gamma_{u,v} = \infty,\ \forall u, v \in [1, |p|+1]$;
5 $Q \leftarrow 1,\ end = 2,\ start = 1$;
6 **while** $start \neq end$ **do**
7 $u = Q(start)$;
8 **for** $v = u+1$ *to* $|p|+1$ **do**
9 invoke *Algorithm* 4 to find the lightpath segment for $u \to v$ on path p, and get the values of $\gamma_{u,v}$, ts and te;
10 **if** $\gamma_{1,v} > \gamma_{1,u} + \gamma_{u,v}$ **then**
11 $\gamma_{1,v} = \gamma_{1,u} + \gamma_{u,v}$;
12 $\hat{t}^s_v = ts,\ \hat{t}^e_v = te$;
13 $X_v = X_u \cup \{(u,v)\}$;
14 $Y_v = Y_u \cup \{(\hat{t}^s_v, \hat{t}^e_v)\}$;
15 **if** $v \notin Q$ **then**
16 $Q \leftarrow v$;
17 $end = end + 1$;
18 **end**
19 **break**;
20 **end**
21 **end**
22 $start = start + 1$;
23 **end**
24 **if** $\gamma_{1,|p|+1} < \infty$ **then**
25 deploy lightpath segments based on $X_{|p|+1}$;
26 serve the DoT and update network status;
27 **break**;
28 **end**
29 **end**

that these network resources are limited in the ML-IDCON, we design a resource-aware lightpath segmentation algorithm to reduce the blocking probability of DoTs. Meanwhile, to limit the time complexity of the lightpath segmentation, we propose a polynomial-time scheme for it by leveraging the branch-and-bound approach.

Algorithm 3 shows the proposed SnF-assisted lightpath segmentation algorithm, which can schedule a DoT on K shortest paths between its source and destination DCs, with the considerations on the extra usages of DC storage and transponders. Similar as that in *Algorithm* 2, we store the K shortest path candidates in $\mathcal{P}_{s_j,d_j}$, and for a path $p \in \mathcal{P}_{s_j,d_j}$, we index the nodes on it as $1, \cdots, |p|+1$. Then, we use a for-loop to check each path in $\mathcal{P}_{s_j,d_j}$ to find a feasible lightpath segmentation scheme for the DoT. *Lines* 2-5 initialize the variables. Here, we use $\{X_u : u \in [1, |p|+1]\}$ to record the lightpath segments from the first node to the u-th one on path p, $\{Y_u : u \in [1, |p|+1]\}$ record the corresponding transfer windows on the lightpath segments, and $\hat{t}^s_u$ and $\hat{t}^e_u$ store the start and end time of the transfer window for the last lightpath segment before reaching node u, respectively. We initialize $\hat{t}^s_u$ and $\hat{t}^e_u$ as the DoT's start time, *i.e.*, t^s_j. $\gamma_{u,v}$ is defined as the cost of the data transfer for $u \to v$, and we initialize it as ∞ before the corresponding lightpaths having been set up. Q is the queue to store reachable nodes on p (*i.e.*, those have lightpath segments to go there) sequentially, and the first node on p (*i.e.*, the source DC) is pushed in Q initially. We use end and $start$ as the variables to assist our operations on Q.

The while-loop covering *Lines* 6-23 checks the nodes on p with the assistance of Q to find a feasible lightpath segmentation scheme. *Line* 7 gets the start node u for the current lightpath segment, and then *Lines* 8-21 check each node v in $[u+1, |p|+1]$ to find the cost-minimized lightpath segment for $u \to v$, and the information of the lightpath segments for $1 \to v$ accordingly. Specifically, *Line* 9 invokes *Algorithm* 4 to find the cost-minimized lightpath segment for $u \to v$ along path p and allocate the corresponding bandwidth, storage and processing resources for it. Then, the information regarding the lightpath segment, *i.e.*, $\gamma_{u,v}$ as the cost of the data transfer for $u \to v$, and (ts, te) as the transfer window on the lightpath segment, where ts and te are the start and end time of data transfer, respectively, is also provided by *Algorithm* 4.

Line 10 compares the previously-known minimum cost $\gamma_{1,v}$ with $\gamma_{1,u} + \gamma_{u,v}$ to check whether an update on the lightpath segmentation scheme would be necessary. Specifically, if $\gamma_{1,u} + \gamma_{u,v}$ yields a smaller cost, we should replace the lightpath segments for X_v with $X_u \cup \{(u,v)\}$ and update other related variables accordingly. And if node v is not in Q, we push it in and update end. When we have checked all the node pairs on p starting from u, *Line* 22 updates $start$. In *Line* 24, if we find $\gamma_{1,|p|+1} < \infty$, which means that we have found a feasible lightpath segmentation scheme for transferring the j-th DoT's data from s_j to d_j along path p, we deploy the lightpath segments according to $X_{|p|+1}$ to serve the DoT, and update the network status, in *Lines* 25-27.

Algorithm 4 shows the detailed procedure to find the cost-minimized lightpath segment for $u \to v$. *Line* 1 initializes the data transfer cost $\gamma_{u,v}$. Then, based on the $\hat{t}^s_u$ and $\hat{t}^e_u$ passed down from *Algorithm* 3, the two for-loops covering *Lines* 2-29 check each feasible transfer window (ts, te) for the DoT. Specifically, *Lines* 4-5 calculate the required bandwidth b_j for the DoT and the residual bandwidth $b^j_{u,v}$ in the IP

layer, respectively. If the residual bandwidth is sufficient (*i.e.*, $b^e_{u,v} \leq b_j$), we can serve the DoT with the bandwidth in the IP layer directly, and thus the cost should be $\gamma_{u,v} = 0$ since no extra DC storage or transponder is required. Otherwise, we need to set up a new lightpath segment to provide the remaining required bandwidth, which can be obtained in the number of FS' as

$$w_j = \lceil \frac{b_j - b^e_{u,v}}{\Delta b_{u,v}} \rceil, \tag{7}$$

where $\Delta b_{u,v}$ is the bandwidth of data transfer from DC $d_p(u)$ to DC $d_p(v)$ along path p when the highest feasible modulation-level is selected. Then, the number of extra transponders for the lightpath segment is

$$N_j = \lceil \frac{w_j}{F} \rceil, \tag{8}$$

based on which we try to find N_j available FS-blocks on the fiber links from $d_p(u)$ to $d_p(v)$ along p with the first-fit scheme. Specifically, the first $N_j - 1$ FS-blocks should have $F + f_g$ spectrally contiguous FS', while the size of the last FS-block is $w_j - (N_j - 1) \cdot F + f_g$ FS'. If these FS-blocks can be found, we define the cost $\gamma_{u,v}$ as

$$\gamma_{u,v} = \sum_{t=\hat{t}^s_j}^{t^e_j} \frac{N_j}{M_{d_p(u)}(t)}, \tag{9}$$

where $\frac{N_j}{M_{d_p(u)}(t)}$ is the ratio of the required transponders to the available transponders on DC $d_p(u)$ in TS t, *i.e.*, the cost $\gamma_{u,v}$ will be higher if we leave fewer transponders on the DC. Otherwise, *Line* 15 stops the current loop and tries to find the feasible FS blocks by changing the transfer window (ts, te).

We calculate the storage space usage on DC $d_p(u)$ in *Lines* 18-23. Here, $S^I_u(t)$ and $S^O_u(t)$ are introduced to represent the cumulative input and output data in DC $d_p(u)$ in TS t, which can be calculated as

$$S^I_u(t) = \begin{cases} \frac{A_j}{\hat{t}^e_u - \hat{t}^s_u + 1} \cdot (t - \hat{t}^s_u + 1), & t \in [\hat{t}^s_u, \hat{t}^e_u], \\ A_j, & t \in [\hat{t}^e_u, te], \end{cases}$$

$$S^O_u(t) = \begin{cases} 0, & t \in [\hat{t}^s_u, ts - 1], \\ \frac{A_j}{\Delta t_j} \cdot (t - ts + 1), & t \in [ts, te]. \end{cases}$$

Hence, the required storage $\hat{S}_u(t)$ in DC $d_p(u)$ is

$$\hat{S}_u(t) = S^I_u(t) - S^O_u(t), \ t \in [\hat{t}^s_u, te]. \tag{10}$$

If the required storage can be accommodated in DC $d_p(u)$, *i.e.*, $\hat{S}_u(t) \leq S_{d_p(u)}(t), \ \forall t$, *Line* 20 updates the cost $\gamma_{u,v}$ as

$$\gamma_{u,v} = \gamma_{u,v} + \sum_{t=\hat{t}^s_u}^{te} \frac{\hat{S}_u(t)}{S_{d_p(u)}(t)}, \tag{11}$$

which also assigns a higher storage cost to the DC if less storage would be left on it. Next, *Lines* 24-27 check whether the processing resources on DC $d_p(u)$ is sufficient if it is the destination DC. Basically, the amount of data that should be processed in TS t is

$$C(t) = \begin{cases} \min\{\frac{C_{d_j}(t)}{\delta_j}, \frac{b_j}{8} + S(t-1)\}, & t \in [ts, te], \\ \min\{\frac{C_{d_j}(t)}{\delta_j}, S(t-1)\}, & t \in [te+1, t^e_j], \end{cases} \tag{12}$$

where $S(t)$ is the amount of data that is stored in destination DC in TS t, which can be obtained as

$$S(t) = \begin{cases} 0, & t = ts - 1, \\ \min\{S(t-1) + \frac{b_j}{8} - \frac{C(t)}{\delta_j}, S_{d_j}(t)\}, & t \in [ts, te], \\ \min\{S(t-1) - \frac{C(t)}{\delta_j}, S_{d_j}(t)\}, & t \in [te+1, t^e_j]. \end{cases} \tag{13}$$

Finally, if the data can be processed in the destination DC (*i.e.*, $\sum_t C(t) \geq A_j$), *Algorithm* 4 returns the result to *Algorithm* 3, since the obtained lightpath segment can transfer the DoT's data from $d_p(u)$ to $d_p(v)$ along p in transfer window (ts, te). Note that, the time complexity of *Algorithm* 4 is $O(D^2_j \cdot (N + D_j))$, while in *Algorithm* 3, each node would be checked at most once in the while-loop. Therefore, the time complexity of *Algorithm* 3 is $O(K \cdot |\mathcal{D}|^2 \cdot D^2_j \cdot (N + D_j))$. Note that, to reduce the time complexity, *Algorithm* 4 uses the first-fit scheme for wavelength/spectrum assignments, and we will consider more sophisticated schemes (*e.g.*, the fragmentation-aware ones in [40, 41]) in our future work.

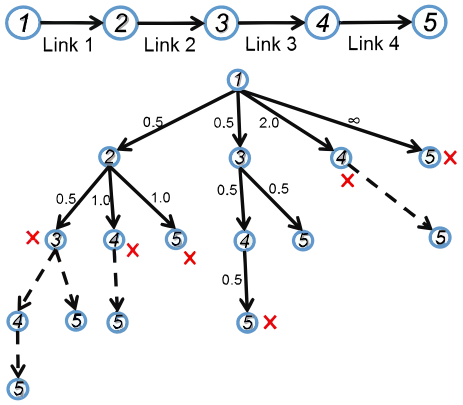


Fig. 3. Example on building lightpath segments with branch-and-bound.

Fig. 3 provides an example on building lightpath segments with the branch-and-bound approach in *Algorithm* 3. The path consists of 5 DCs and 4 fiber links. We first try to find the lightpath segments from node 1 to nodes $\{2, 3, 4, 5\}$ by applying *Algorithm* 4 repeatedly for the corresponding node pairs, and the obtained costs $\{\gamma_{u,v}\}$ are assumed to be $0.5, 0.5, 2.0$ and ∞, respectively. Then, we proceed to check the lightpath segments from node 2, and find that $\gamma_{1,3} < \gamma_{1,2} + \gamma_{2,3}$. Hence, the branch 1→2→3 should be pruned. By repeating these operations, we can eventually find the lightpath segment 1→3→5, whose cost is the minimum.

Algorithm 4: Find the Cost-Minimized Lightpath Segment for $u \rightarrow v$

$\gamma_{u,v} = \infty$;
for $ts = \hat{t}_u^s$ *to* t_j^e **do**
  for $te = \max(\hat{t}_u^e, ts)$ *to* t_j^e **do**
    $b_j = \frac{8 \cdot A_j}{(te - ts + 1) \cdot \Delta t}$;
    $b_{u,v}^e = \min\left(\{b_{d_p(u),d_p(v)}(t) : t \in [ts, te]\}\right)$;
    if $b_{u,v}^e \geq b_j$ **then**
      $\gamma_{u,v} = 0$;
    else
      select the highest feasible modulation -level and get $\Delta b_{u,v}$;
      get w_j and N_j with Eqs. (7) and (8);
      find N_j available FS-blocks on links from $d_p(u)$ to $d_p(v)$ along p with first-fit;
      if *the FS-blocks can be found* **then**
        $\gamma_{u,v} = \sum_{t=t_j^s}^{t_j^e} \frac{N_j}{M_{d_p(u)}(t)}$;
      else
        continue;
      end
    end
    calculate $\{\hat{S}_u(t)\}$ with Eq. (10);
    if $\hat{S}_u(t) \leq S_{d_p(u)}(t),\ \forall t$ **then**
      $\gamma_{u,v} = \gamma_{u,v} + \sum_{t=\hat{t}_u^s}^{te} \frac{\hat{S}_u(t)}{S_{d_p(u)}(t)}$;
    else
      continue;
    end
    calculate $\{C(t)\}$ and $\{S(t)\}$ with Eqs. (12) and (13) iteratively;
    if $\sum_{t=ts}^{t_j^e} C(t) \geq A_j$ **then**
      return $\gamma_{u,v},\ ts,\ te$;
    end
  end
end
return $\gamma_{u,v},\ ts,\ te$;

D. Modifications for Fixed-Grid ML-IDCONs

Up to now, the algorithms are designed for flexible-grid ML-IDCONs. For a fixed-grid ML-IDCON, we have $f_g = 0$ since there is no need to allocate a guard-band for each lightpath, and as one transponder covers a single wavelength channel, we set $F = 1$ in the fixed-grid ML-IDCON. Then, the operations in *Algorithms* 2-3 can be directly leveraged to schedule DoTs in a fixed-grid ML-IDCON.

V. Performance Evaluations

In this section, we perform numerical simulations to compare the performance of fixed- and flexible-grid ML-IDCONs on DoT scheduling, and also benchmark the performance of our proposed algorithm with an existing one.

A. Simulation Setup

We assume that the ML-IDCON uses NSFNET as its topology, which includes 14 nodes and 21 bi-directional fiber links [42]. Each node contains a DC, whose computing and storage resources (*i.e.*, $C_i(t)$ and $S_i(t)$) are randomly selected within $[500, 1000]$ units and $[50, 100]$ TB, respectively. Considering the time required for lightpath reconfiguration, we set a TS as 5 minutes [43] and make each simulation last for 200 TS', *i.e.*, $T = 200$ TS'. We assume that each fiber link has 4 THz available spectrum. Hence, for the fixed-grid ML-IDCON, each link can accommodate 80 wavelength channels, each of which is 50 GHz. Two types of single line-rate (SLRs), which provide channel capacities of 40 and 100 Gb/s, respectively, and one type of mixed line-rate (MLR) which supports 10, 40 and 100 Gb/s channel capacities, are considered. On the other hand, the flexible-grid ML-IDCON has 320 FS' on each link, each of which is 12.5 GHz. The flexible-grid ML-IDCON supports four modulation-levels, *i.e.*, BPSK, QPSK, 8QAM and 16QAM, which can provide 12.5, 25, 37.5 and 50 Gb/s capacity with an FS, respectively. Each BV-T can groom up to $F = 8$ FS', and each lightpath requires $f_g = 1$ FS as the guard-band. Here, we use the models reported in [36] to obtain the transmission reach of optical signals and the power consumption of transponders, which are summarized in Table I. In addition, the power consumption of an IP router port is assumed to be 10 W/Gbps [44, 45].

TABLE I
Simulation setting of transponders [36]

Transponder Type		Distance (km)	Power Consumption (W)
Flexible-grid	BPSK	4000	112.4
	QPSK	2000	133.4
	8QAM	1000	154.5
	16QAM	500	175.5
Fixed-grid	10 Gb/s	3200	34.1
	40 Gb/s	2200	98.9
	100 Gb/s	1800	351.0

The performance evaluations compare our proposed algorithm (OURS) with three benchmark algorithms.

- *SSD*: We design the benchmark SSD to follow the general procedure of OURS but not consider the SnF scheme, *i.e.*, only the storage space in the source and destination DCs can be used for DoT scheduling. This benchmark is used to explore how much performance improvement that our proposal can achieve over the schemes that only consider intermediate O/E/O conversions for DoT scheduling in ML-IDCONs, *i.e.*, to confirm the effectiveness of SnF. Note that, although the algorithms designed in [7, 17, 23] were also for DoT scheduling, their network models and problem formulations are different from those in this work. Hence, we cannot use them as benchmarks.
- *tDGA+*: This algorithm is modified from the tDGA+ algorithm in [19], which is the best-known existing scheme that also considers SnF-assisted lightpath segmentation for DoT scheduling in ML-IDCONs. We extend it with the spectrum allocation scheme of OURS to handle the DoT scheduling in flexible-grid ML-IDCONs. Also, as the tDGA+ in [19] uses the time-division multiplexing (TDM) model in data transfer, we set the duration of a frame as 1 TS and let each frame contain 5 slots in the simulations of tDGA+, same as the settings in [19].

- *GREEDY*: This is a straightforward benchmark that greedily allocates resources to serve each DoT in the sorted order. Specifically, it first tries to allocate bandwidth on the K shortest paths between the DoT's source and destination in the IP layer, and if the transfer throughput in the IP layer is not enough, it will build a lightpath for the DoT with K-shortest path routing and first-fit spectrum assignment using the highest feasible modulation-level.

Each simulation generates DoTs dynamically and the number of DoTs per TS (*i.e.*, $\hat{J}$) follows the uniform distribution within $[10, 100]$. The source and destination DCs of each DoT are chosen randomly. The data amount of each DoT, *i.e.*, A_j, is uniformly distributed within $[500, 5000]$ GB, its maximum sojourn time D_j is set within $[1, 10]$ TS', and the computing resource that is required to process 1 TB data for it is selected within $[1, 5]$ units. Meanwhile, we randomly generate flow-oriented lightpaths in the ML-IDCONs as the background traffic, which occupies $\eta = 0.5$ of the total spectra. For the DoT scheduling, we pre-calculate $K = 3$ shortest routing paths for each node pair in the topology. To obtain each data point, we perform 10 independent simulations and average the results to ensure sufficient statistical accuracy. Table II summarizes the key simulation parameters.

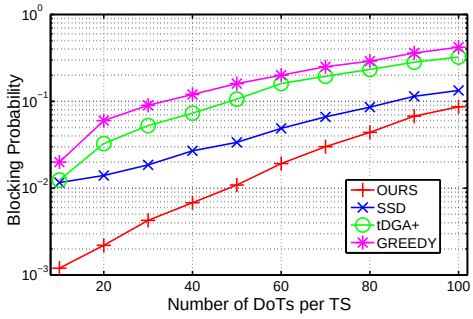
TABLE II
KEY SIMULATION PARAMETERS

$\hat{J}$, number of DoTs per TS	[10,100]
A_j, data size of a DoT	[500, 5000] GB
D_j, maximum sojourn time of a DoT	[1, 10] TS
η, ratio of spectrum usage of background traffic	0.5
K, number of shortest paths	3
$C_i(t)$, computing resources in DC i in TS t	[500, 1000] units
$S_i(t)$, storage in DC i in TS t	[50, 100] TB
time of a TS	5 minites
number of TS' of each simulation	200

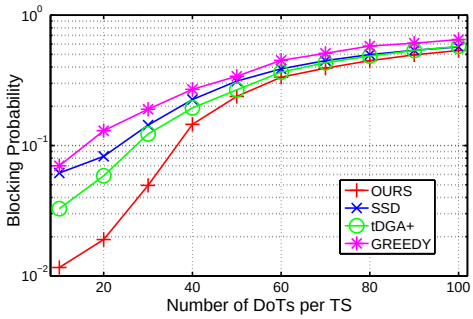
B. Simulation Results and Analysis

We use the algorithms to schedule the DoTs in fixed- and flexible-grid ML-IDCONs and compare their performance in terms of blocking probability, transponder power consumption, and DC storage usage.

1) Impacts on Blocking Probability: Figs. 4 and 5 show the simulation results on blocking probability from the algorithms in different ML-IDCONs. Firstly, we can see that OURS achieves the lowest blocking probability in both the fixed- and flexible-grid ML-IDCONs. And for the same algorithm and arrival rate of the DoTs, the blocking probability of the flexible-grid ML-IDCON is much lower than that of the fixed-grid one. This suggests that flexibility in the optical layer does benefit the service provisioning of DoTs. By comparing the results in Fig. 5, we observe that the blocking probability of SLR 100 Gb/s is higher than that of SLR 40 Gb/s, when the DoT arrival rate is relatively small. This is because the transmission reach of 100 Gb/s transponders is short, and thus certain K shortest path candidates might not be feasible for 100 Gb/s signals even though they still have available spectra. When the DoT arrival rate continues to increase, the blocking probability of SLR 40 Gb/s soon becomes the highest. This is because its spectrum efficiency is the lowest.



(a) Flex-grid based ML-IDCON



(b) MLR based ML-IDCONs

Fig. 4. Blocking probability in flex-grid and MLR based ML-IDCONs.

2) Impacts on Transponder Power Consumption: The results on average energy used by per GB data from the algorithms in different ML-IDCONs are listed in Table III. Here, we use $\hat{J}$ to denote the arrival rate of DoTs. We list the results from OURS with their absolute values, while those from benchmarks are listed as their relative ratios to the corresponding results from OURS. It is promising to notice that the flexible-grid ML-IDCON also achieves the lowest power consumption when a specific algorithm is used. This is because the flexible-grid ML-IDCON can set up lightpaths adaptively according to the exact bandwidth demands of the DoTs, *i.e.*, avoiding unnecessary power consumption on the transponders. As for the fixed-grid ML-IDCONs, the power consumption of SLR 100 Gb/s is the highest. Meanwhile, we notice that OURS consumes slightly more power than the benchmarks. This is due to the fact that OURS tries to utilize the storage space on intermediate DCs the best to build SnF-assisted lightpaths for minimizing DoT blocking.

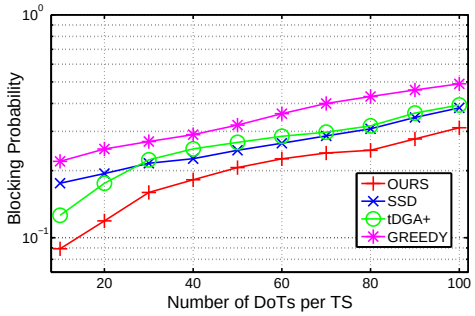
3) Impacts on Storage Resource: Table IV shows the results on the average storage used by per GB data from different algorithms, and it lists the results in the same way as that in Table III. We observe that when a specific algorithm is used, the average storage usage of the flexible-grid ML-IDCON is still the smallest. Moreover, we notice that in the flexible-grid ML-IDCON, only less than 0.02 GB storage space is required on average for OURS to schedule 1 GB data transfer when the DoT arrival rate is the highest. This verifies that OURS uses very little DC storage in the DoT scheduling. For the fixed-grid ML-IDCONs, the DC storage usage of SLR 40 Gb/s increases the fastest with the DoT arrival rate. This is

TABLE III
AVERAGE ENERGY USED BY PER GB DATA

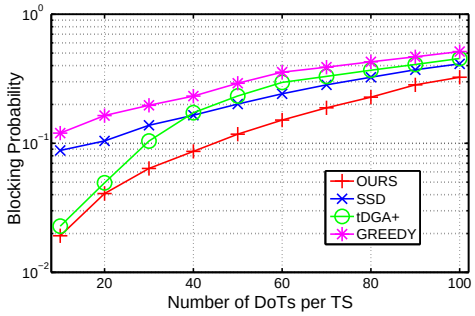
		Flex-grid			SLR 40 Gb/s			SLR 100 Gb/s			MLR		
$\hat{J}$		30	60	90	30	60	90	30	60	90	30	60	90
Absolute Value	*OURS* (J)	183.4	189.5	192.2	227.6	224.5	219.6	275.7	269.1	264.6	254.9	254.1	251.0
Relative Ratio to *OURS*	*SSD*	0.97	0.96	0.94	0.96	0.94	0.93	0.96	0.96	0.95	0.96	0.96	0.93
	tDGA+	0.89	0.89	0.87	0.96	0.94	0.93	0.95	0.97	0.9711	0.90	0.91	0.92
	GREEDY	0.81	0.80	0.80	0.96	0.94	0.93	0.86	0.89	0.92	0.82	0.83	0.86

TABLE IV
AVERAGE STORAGE USED BY PER GB DATA

		Flex-grid			SLR 40 Gb/s			SLR 100 Gb/s			MLR		
$\hat{J}$		30	60	90	30	60	90	30	60	90	30	60	90
Absolute Value	*OURS* (GB)	0.004	0.008	0.017	0.018	0.063	0.077	0.012	0.017	0.027	0.010	0.023	0.041
Relative Ratio to *OURS*	*SSD*	1.18	1.08	0.99	0.56	0.48	0.41	0.44	0.55	0.51	0.56	0.48	0.41
	tDGA+	12.06	6.08	3.06	4.77	2.43	1.44	2.59	1.79	1.20	4.77	2.43	1.44
	GREEDY	0.42	0.32	0.25	0.25	0.26	0.21	0.33	0.37	0.36	0.25	0.26	0.21



(a) SLR 40 Gb/s based ML-IDCONs



(b) SLR 100 Gb/s based ML-IDCONs

Fig. 5. Blocking probability in SLR based ML-IDCONs.

because without 100 Gb/s channels, the data transfer capacity of SLR 40 Gb/s might not be sufficient for the DoTs, and thus the SnF scheme would be used most frequently. In a specific ML-IDCON, OURS uses more storage space than GREEDY and SSD since they do not consider SnF, but it consumes significantly less storage than tDGA+. Hence, the results verify that the lightpath segmentation scheme for SnF in OURS is more intelligent and cost-efficient than that in tDGA+.

In all, after analyzing the results in Figs. 4 and 5 and Tables III and IV, we can conclude that for DoT scheduling, the flexible-grid ML-IDCON outperforms the fixed-grid ones in blocking probability, transponder power consumption, and DC storage usage. Meanwhile, OURS achieves the lowest blocking probability in both the fixed- and flexible-grid ML-IDCONs.

4) Average Running Time: Table V summarizes the algorithms' average running time to schedule a DoT. All the simulations are based on Matlab and run on a computer with a 3.30 GHz Intel I3-2120 CPU and 8 GB memory. We can see that for OURS, SSD and GREEDY, the running time only increases slightly with $\hat{J}$, while the running time of tDGA+ is significantly longer and increases more rapidly. This is because the lightpath segmentation scheme in tDGA+ is not time-efficient. As OURS considers SnF in DoT scheduling, it takes slightly longer running time than SSD and GREEDY. However, as OURS only takes a few milliseconds to serve a DoT, it fits into the requirement of dynamic provisioning well.

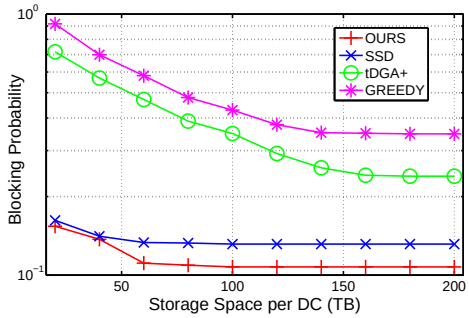
TABLE V
AVERAGE RUNNING TIME TO SERVE A DOT (MSEC)

	$\hat{J}$	Flex-grid	SLR 40 Gb/s	SLR 100 Gb/s	MLR
OURS	30	6.18	4.28	4.06	6.05
	60	7.36	4.91	4.45	6.49
	90	7.82	5.06	4.48	6.71
SSD	30	3.96	3.52	2.99	4.24
	60	4.36	3.66	3.22	4.83
	90	5.48	3.80	3.52	4.87
tDGA+	30	18.89	16.43	15.28	20.97
	60	24.54	20.55	22.51	31.76
	90	27.20	24.80	24.28	35.55
GREEDY	30	2.93	2.74	2.52	3.45
	60	3.46	3.20	2.94	3.86
	90	4.02	3.47	3.43	4.55

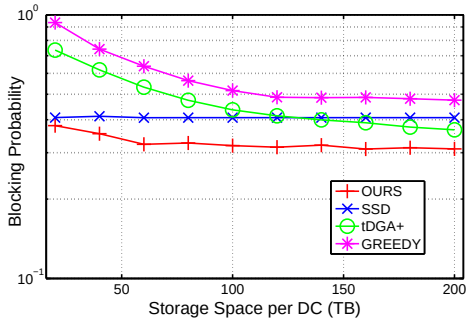
C. Impacts of Simulation Parameters

Finally, we investigate the simulation parameters' impacts on the algorithms' performance. We first fix the DoT arrival rate as $\hat{J} = 100$ per TS and change the storage space in each DC. Fig. 6 shows the results on blocking probability from the algorithms when the storage space per DC is different. It is clear that in any ML-IDCON, the blocking probability from OURS is still the lowest. Meanwhile, when a same algorithm is used, the blocking probability in the flexible-grid based ML-IDCON is much lower than that in the MLR based one. It

is interesting to notice that for all the simulation scenarios, the blocking probability first decreases with the storage space per DC, and then converges to a fixed value. This is because when the DC storage space is relatively small, it could be the bottleneck factor to cause DoT blocking. When it gets increased to certain point, it would no longer be the bottleneck, and thus providing more DC storage would not decrease the blocking probability any more. Compared with tDGA+, OURS require less storage space per DC to reach the steady point. For instance, in the flexible-grid ML-IDCON, OURS only requires ~ 60 TB storage per DC to reach the steady point in Fig. 6(a), while tDGA+ needs at least 160 TB storage per DC to make its blocking probability converge.



(a) Flex-grid based ML-IDCON



(b) MLR based ML-IDCON

Fig. 6. Blocking probability under different storage space per DC.

Then, we analyze the impact of the background traffic, still by fixing the DoT arrival rate as $\hat{J} = 100$ per TS. Fig. 7 shows the blocking probability results when the background traffic ratio η changes within $[0.3, 0.7]$. We can still see that OURS performs the best in terms of blocking probability, and when a same algorithm is used, the blocking probability in the flexible-grid based ML-IDCON is lower than that in the MLR based one. We also study the impact of the routing paths' hop-count on the blocking probability. Specifically, when generating the DoTs, we fix the average hop-count of the shortest paths between their source and destination DCs, and then run the simulations. Here, we denote the average minimum hop-count of the DoTs as H, and Fig. 8 shows the blocking probability results when we have $\hat{J} = 100$ and $\eta = 0.5$. It can be seen that the blocking probability generally increases with H in all the simulation scenarios, which is because when H is larger, it will become more difficult for the algorithms to serve the DoTs. Meanwhile, the results still confirm that OURS outperforms all the benchmarks in terms of blocking probability, and the

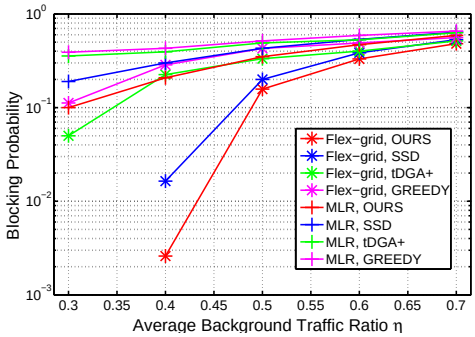


Fig. 7. Blocking probability under different background traffic ratios.

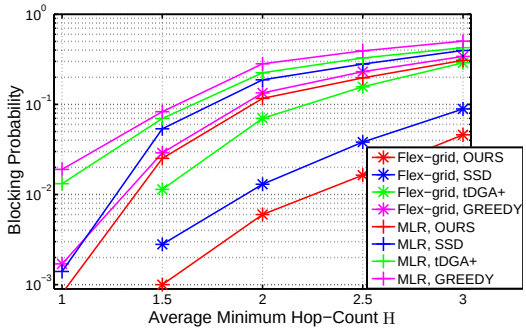


Fig. 8. Blocking probability under different average minimum hop-counts.

blocking probability in the flexible-grid based ML-IDCON is lower than that in the MLR based one.

VI. Conclusion

In this paper, we conducted a comparison study on the scheduling of DoTs in fixed- and flexible-grid ML-IDCONs. We proposed a DoT scheduling algorithm that can work well for both of them. For each DoT, the algorithm first tried to serve it using the residual bandwidth in the IP layer. To achieve this, we leveraged the TEN approach and transformed the SnF-assisted DoT scheduling problem into an MCMF problem. Then, if the obtained data transfer throughput was not sufficient for the DoT, the algorithm tried to build lightpath segments for it by using the branch and bound scenario. We performed extensive simulations to evaluate the proposed algorithm's performance in fixed- and flexible-grid ML-IDCONs, and also compared it with three benchmarks. Simulation results indicated that for DoT scheduling, the flexible-grid ML-IDCON could outperform fixed-grid ones in terms of the blocking probability, energy consumption of transponders, and usage of DC storage, and our algorithm achieved lower blocking probability than the benchmarks with comparable or higher time-efficiency.

Acknowledgments

This work was supported in part by the NSFC Project 61371117, the Key Project of the CAS (QYZDY-SSW-JSC003), and the NGBWMCN Key Project under Grant No. 2017ZX03001019-004.

REFERENCES

[1] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[2] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *Proc. of SIGCOMM 2013*, pp. 3–14, Aug. 2013.

[3] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Trans. Multimedia*, vol. 17, pp. 1297–1308, Aug. 2015.

[4] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[5] N. Bitar, S. Gringeri, and T. Xia, "Technologies and protocols for data center and cloud networking," *IEEE Commun. Mag.*, vol. 51, pp. 24–31, Sept. 2013.

[6] S. Liu, W. Lu, and Z. Zhu, "On the cross-layer orchestration to address IP router outages with cost-efficient multilayer restoration in IP-over-EONs," *J. Opt. Commun. Netw., in Press*, 2017.

[7] A. Patel and J. Jue, "Routing and scheduling for variable bandwidth advance reservation," *J. Opt. Commun. Netw.*, vol. 3, pp. 912–923, Dec. 2011.

[8] W. Lu and Z. Zhu, "Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 2078–2086, May. 2015.

[9] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with Netstitcher," in *Proc. of SIGCOMM 2011*, pp. 74–85, Aug. 2011.

[10] X. Xie, Q. Ling, P. Lu, and Z. Zhu, "ADMM-based distributed algorithm for emergency backup in time-variant inter-DC networks," in *Proc. of ICC 2017*, pp. 1–6, May 2017.

[11] W. Lu *et al.*, "Dynamic multi-path service provisioning under differential delay constraint in elastic optical networks," *IEEE Commun. Lett.*, vol. 17, pp. 158–161, Jan. 2013.

[12] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Time-varying spectrum allocation policies and blocking analysis in flexible optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, pp. 13–25, Jan. 2013.

[13] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.*, vol. 31, pp. 1621–1627, May 2013.

[14] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[15] W. Lu, Z. Zhu, and B. Mukherjee, "On hybrid IR and AR service provisioning in elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 4659–4669, Nov. 2015.

[16] H. Wu, F. Zhou, Z. Zhu, and Y. Chen, "On the distance spectrum assignment in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 2391–2404, Aug. 2017.

[17] J. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in geo-distributed optical inter-datacenter networks," *J. Lightw. Technol.*, vol. 33, pp. 3005–3015, Jul. 2015.

[18] W. Lu, Z. Zhu, and B. Mukherjee, "Optimizing deadline-driven bulk-data transfer to revitalize spectrum fragments in EONs," *J. Opt. Commun. Netw.*, vol. 7, pp. B173–B183, Dec. 2015.

[19] D. Feng, W. Sun, and W. Hu, "Joint provisioning of lightpaths and storage in store-and-transfer wavelength-division multiplexing networks," *J. Opt. Commun. Netw.*, vol. 9, pp. 218–233, Mar. 2017.

[20] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[21] W. Shi, Z. Zhu, M. Zhang, and N. Ansari, "On the effect of bandwidth fragmentation on blocking probability in elastic optical networks," *IEEE Trans. Commun.*, vol. 61, pp. 2970–2978, Jul. 2013.

[22] Z. Zhu *et al.*, "Impairment- and splitting-aware cloud-ready multicast provisioning in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 1220–1234, Apr. 2017.

[23] C. Joe-Wong, I. Kamitsos, and S. Ha, "Interdatacenter job routing and scheduling with variable costs and deadlines," *IEEE Trans. Smart Grid*, vol. 6, pp. 2669–2680, Nov. 2015.

[24] Q. Pu *et al.*, "Low latency geo-distributed data analytics," in *Proc. of SIGCOMM 2015*, pp. 421–434, Aug. 2015.

[25] Y. Wang, S. Su, A. Liu, and Z. Zhang, "Multiple bulk data transfers scheduling among datacenters," *Comput. Netw.*, vol. 68, pp. 123 – 137, Aug. 2014.

[26] P. Lu, K. Wu, Q. Sun, and Z. Zhu, "Toward online profit-driven scheduling of inter-DC data-transfers for cloud applications," in *Proc. of ICC 2015*, pp. 5583–5588, Jun. 2015.

[27] K. Wu, P. Lu, and Z. Zhu, "Distributed online scheduling and routing of multicast-oriented tasks for profit-driven cloud computing," *IEEE Commun. Lett.*, vol. 20, pp. 684–687, Apr. 2016.

[28] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[29] D. Andrei *et al.*, "Provisioning of deadline-driven requests with flexible transmission rates in WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 353–366, Apr. 2010.

[30] P. Yi, H. Ding, and B. Ramamurthy, "Budget-optimized network-aware joint resource allocation in grids/clouds over optical networks," *J. Lightw. Technol.*, vol. 34, pp. 3890–3900, Aug. 2016.

[31] Z. Zhu *et al.*, "Energy-efficient translucent optical transport networks with mixed regenerator placement," *J. Lightw. Technol.*, vol. 30, pp. 3147–3156, Oct. 2012.

[32] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, pp. 1354–1366, May 2011.

[33] E. Palkopoulou *et al.*, "Quantifying spectrum, cost, and energy efficiency in fixed-grid and flex-grid networks," *J. Opt. Commun. Netw.*, vol. 4, pp. B42–B51, Nov. 2012.

[34] J. Yin *et al.*, "Experimental demonstration of building and operating QoS-aware survivable vSD-EONs with transparent resiliency," *Opt. Express*, vol. 25, pp. 15 468–15 480, 2017.

[35] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, pp. 1520–1523, Sept. 2012.

[36] J. Lopez *et al.*, "On the energy efficiency of survivable optical transport networks with flexible-grid," in *Proc. of ECOC 2012*, pp. 1–3, Sept. 2012.

[37] V. Curri, M. Cantono, and R. Gaudino, "Elastic all-optical networks: a new paradigm enabled by the physical layer. how to optimize network performances?" in *Proc. of ECOC 2016*, pp. 1–3, Sept. 2016.

[38] W. Fang *et al.*, "Joint defragmentation of optical spectrum and IT resources in elastic optical datacenter interconnections," *J. Opt. Commun. Netw.*, vol. 7, pp. 314–324, Mar. 2015.

[39] J. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Opera. Res.*, vol. 41, pp. 338–350, Mar./Apr. 1993.

[40] F. Cugini *et al.*, "Push-pull defragmentation without traffic disruption in flexible grid optical networks," *J. Lightw. Technol.*, vol. 31, pp. 125–133, Jan. 2013.

[41] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightw. Technol.*, vol. 32, pp. 1014–1023, Mar. 2014.

[42] M. Zhang, C. You, and Z. Zhu, "On the parallelization of spectrum defragmentation reconfigurations in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 2819–2833, Oct. 2016.

[43] X. Jin *et al.*, "Optimizing bulk transfers with software-defined optical WAN," in *Proc. of SIGCOMM 2016*, pp. 87–100, Aug. 2016.

[44] Y. Kim, C. Lee, J. Rhee, and S. Lee, "IP-over-WDM cross-layer design for green optical networking with energy proportionality consideration," *J. Lightw. Technol.*, vol. 30, pp. 2088–2096, Jul. 2012.

[45] W. Heddeghem *et al.*, "Power consumption modeling in optical multilayer networks," *Photonic Netw. Commun.*, vol. 24, pp. 86–102, Oct. 2012.