JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 109-125 (2020)
DOI: 10.6688/JISE.202001_36(1).0007

Delay-Variation Constrained Spectrum Extraction and Contraction Problem for Multipath Routing on Elastic Optical Networks

DER-RONG DIN
Department of Computer Science and Information Engineering
National Changhua University of Education
Changhua City, 500 Taiwan
E-mail: deron@cc.ncue.edu.tw

The bandwidth requirement of an end-to-end request between the source and destination nodes varies dynamically with time (denoted as *time-varying traffic*). For serving time-varying traffic on an elastic optical network (EON), the frequency spectrum allocated for the request can be expanded or contracted to meet the bandwidth requirement. Multipath routing can reduce the blocking probability of requests for EONs, but the delay-variation between these lightpaths should be considered when establishes these lightpaths. In this paper, the *Delay-Variation Constrained Spectrum Expansion and Contraction Problem* (DVC-SECP) for multipath routing on EONs is studied with time-varying traffic. The expansion/-contraction algorithms and several path-selecting policies (PSPs) are proposed to solve this problem. Simulations show that the proposed algorithms can achieve good results.

Keywords: spectrum expansion and contraction, delay-variation, elastic optical network, multi-path routing, time-varying traffic

1. INTRODUCTION

Elastic optical networks (EONs), which employ *optical-orthogonal frequency division multiplexing* (O-OFDM) technology, have been proposed to scale the demands by efficiently utilizing the spectrum as they provide finer spectrum granularity and distance adaptive modulation formatting. The spectrum of a link (or fiber) in EONs is divided into small unit *frequency slots* (FSs) and the necessary amount of consecutive FSs are assigned to support the request. Besides, more efficient spectrum allocation is achieved in these networks due to flexible grid and elastic line rates providing finer granularity [1, 2]. EONs also provide a super-channel connectivity for accommodating ultra-high capacity demands and a sub-wavelength granularity for low-rate transmissions [1, 2]. Hence, O-OFDM can achieve subwavelength granularity, by using elastic bandwidth (BW) allocation that manipulates the FSs. Specifically, a BW-variable O-OFDM transponder can assign an appropriate number of FSs to serve a request using just-enough FSs.

In an optical network, each optical connection can be transmitted by an allocated channel which consists a *central frequency* (CF) and a *size*. The size of the channel is determined by the requested bit-rate, the modulation technique applied, the (fixed) slice width and the guard band (GB) introduced to separate two spectrum adjacent connections, among others. Due to the *spectrum continuity constraint* [1, 2], the *routing and spectrum assignment* (RSA) has emerged as the essential problem for spectrum management on

Received January 8, 2018; revised April 1, 2018; accepted April 22, 2018.
Communicated by Xiaohong Jiang.
* This work was supported in part by the MOST project under Grant 105-2221-E-018-018.

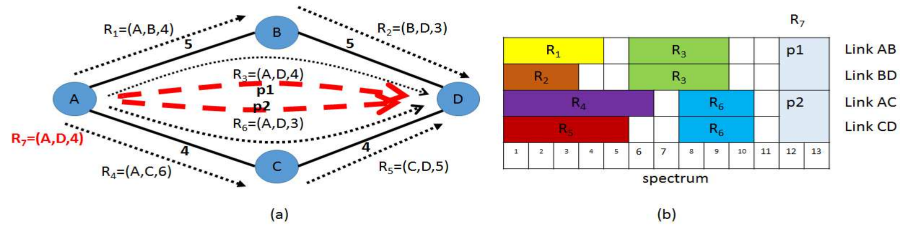

Fig. 1. (a) Multipath routing for $R_7 = (A, D, 4)$; (b) Spectrum allocation for lightpaths $p1$ and $p2$ [3].

EONs. A request requiring a certain capacity should be satisfied by assigning a number of contiguous FSs. For a given request, the goal of the RSA problem is to find a lightpath on the network and assign the number of required FSs.

1.1 Multipath Routing

Multi-path routing scheme has already demonstrated to improve network performance in Wavelength Division Multiplexing (WDM) networks [3]. For online provisioning, it is difficult to serve certain large bandwidth requests with single-path routing due to the bandwidth limitation, thus resulting in high request blocking probability [3, 4]. The EON enables us to split a request's traffic over multiple routing lightpaths without causing significant bandwidth waste. In [5], Lu *et al.* proposed a dynamic multi-path service-provisioning algorithm that is specifically designed for EONs and considers the differential delay constraint. In Fig. 1 (a), the request $R_7 = (A, D, 4)$ between nodes A and D required four FSs can be achieved by two lightpaths $p1$ and $p2$. The allocated FSs for these lightpaths are shown in Fig. 1 (b) [3]. If multipath routing were not allowed, the request might be blocked.

In [4], Zhu *et al.* proposed two dynamic service provisioning algorithms that incorporate a hybrid single-/multi-path routing scheme. The routing model considered in [4] using the routing, modulation level and spectrum assignment (RMSA) model, in which, the routing path, the distance-adaptive modulation format and the allocated FSs of the request should be determined accordingly.

1.2 Multipath Routing with Delay-variation Constraint

In [6], the *virtual topology design* (VTD) problem on EONs was considered. Given the physical network and the traffic demand matrix, the goal of the VTD problem is to find the routing paths and the allocated FSs of the demand of each pair of nodes such that the total cost of transponders can be minimized. In [6], multipath routing was allowed and the delay-variation constraint between lightpaths for same demand was considered. The routing model considered in [6] is the RSA model, but with the delay-variation constraint. Two heuristic algorithms were proposed in [6], they are *Minimum Delay Path First* (MDPF) and *Maximal Allocates First* (MAF).

1.3 Time-Varying Traffic

Several *spectrum allocation* schemes that change the bandwidth of connection dynamically have been studied in [7–11]. In [7] authors used two connections adjacent to share the optical spectra. A general policy to allocate FSs to time-varying traffic was presented in [8–11]. In [8], authors defined a general spectrum allocation framework for time-varying traffic demands on EONs. They discerned three SA schemes (fixed, semi-elastic, and elastic) of different levels of elasticity. For the elastic scheme, both the assigned CF and the size can be subject to change by performing Spectrum Expansion/-Contraction (SEC) in each time interval. In [8, 9], simulations showed that the elastic scheme with expansion/contraction can minimize the amount of unserved bit-rate. Since the performance tradeoff of this scheme is low, spectrum Expansion/Contraction can be considered as an attractive approach for elastic spectrum assignment.

In [12, 13], several SEC schemes are proposed. Furthermore, the EONs enable to expand/contract slot width of channel [14]. Future EONs will change slot width according to time-varying traffic by changing the number of FSs flexibility. In [15], the SEC problem for the multipath routing scheme for RMSA model without delay-variation constraint on EONs was studied. In [16], authors studied the traffic grooming problem in EONs for time-varying traffic. When a request arrives, the control plan determines how to route the request through a combination of new lightpath and existing lightpath which can accommodate the new request by allocating more subcarriers. In [16], the number of allocated FSs of the existing lightpath was not changed.

1.4 Motivation and Contribution

In the paper, the elastic scheme is used, in which, both the CF and the size of the allocated channel can be subject to change by performing Spectrum Expansion/Contraction in each time interval. The *Delay-Variation Constrained Spectrum Expansion/Contraction Problem* (DVC-SECP) for multipath routing on EONs with time-varying traffic is studied. In DVC-SECP, the new request is served by updating (deleting expanding or contracting) currently deployed lightpaths which are found by performing the static algorithms proposed in [6]. The RSA model is considered and the delay-variation constraint is included to constrain the multipath routing, which is not considered in the SECP with RMSA model [15]. To the best of our knowledge, this is the first paper that focuses on the SECP for multipath routing and delay-variation constraint in EONs.

In this paper, several heuristic algorithms (Path-Removing, Expansion, Path Adding and Contraction) are proposed to perform the traffic update. To select lightpaths for updating, several path-selecting policies (PSPs) are designed for each algorithm. Two path adding algorithms (PAAs) are proposed to find DVC-satisfy routing lightpaths. The performance of the proposed algorithms is examined through simulations.

The rest of the paper is organized as follows. First, in Section 2 the definition and assumptions of the problem are given. In Section 3 the proposed algorithms are described (includes SEC algorithm and the PSPs. Then, in Section 4, the performance of the proposed methods is examined. The conclusion is drawn in Section 5.

2. PROBLEM DEFINITION

In this paper, the DVC-SECP on EONs is studied. When a new request arrives, the multipath RSA algorithm (MDPF or MAF in [6] is performed to find a set of lightpaths which satisfies the delay-variation constraint and allocate required FSs to these lightpaths. If the required FSs cannot be allocated or the DVC cannot be satisfied, then the request is blocked. If the request is an adjusted request (that is, there exists one or several lightpaths with the same source and destination nodes), based on the selected SEC policy, these lightpaths are adjusted. In this paper, both the CF and the size of each lightpath of the current request can be adjusted (deleted, expanded or contracted). If the bandwidth variation can be accommodated and the DVC can be satisfied, then the adjusted request is accepted. Otherwise, the multipath RSA algorithm is performed to route the extra bandwidth requirement (if needed).

It is worth noting that the value of DVC of the adjusted request can be changed in this paper. If the value of DVC increases, the currently deployed lightpaths still can be used to route the request, the allocated resources are adjusted to meet the requirement. Otherwise, some lightpaths should be deleted, some lightpaths should be contracted, or some new lightpaths should be added to route the adjusted request. Consider the example shown in Fig. 1 (a), two lightpaths $p1$ and $p2$ with DVC 2.0 are used to route the request $R_7(A,D,4FSs,2.0)$. The adjusted request $R_7(A,D,1FS,2.0)$ means that the bandwidth of the request R_7 is decreased from 4 to 1 FSs and DVC is unchanged. Thus, for the adjusted request, one of the lightpaths ($p1$ and $p2$) should be deleted and the other lightpath should be contracted. For the adjusted request $R_7(A,D,5FSs,2.0)$, the bandwidth of the request R_7 is increased from 4 to 5 FSs. The bandwidth of one of the lightpaths should be increased to route the request if possible. Obviously, only the allocated FSs of path p1 can be increased in Fig. 1 (b).

In this paper, the DVC-SECP for multipath routing with time-varying traffic on EONs is studied. For a given EON and a sequence of adjusted requests, the goal is to add/delete/expand/contract lightpaths and assigned suitable channels to the lightpaths to meet the traffic requirement and delay-variation constraint such that the performance measure can be optimized. In the following, the assumptions, constraints, notations, and definitions of the studied problem are given.

2.1 Assumptions

The assumptions of the DVC-SECP on EONs are given as follows. (1) For each link, there is a fiber connecting the end-nodes and signal can be transmitted bidirectionally. (2) All nodes in the network are equipped with *bandwidth variable wavelength cross-connects* [1, 2]. (3) For simplicity, the numbers of FSs provided by fibers are all equal. (4) The bandwidth requirement between nodes can be transmitted by using multiple lightpaths with same or different routes and numbers of FSs. (5) A guard band (GB) should be allocated between two lightpaths. (6) The number of FSs that a transceiver or switching node can support should be less than or equal to F.

Four constraints are considered in this paper, they are *spectrum continuity constraint*, *subcarrier consecutiveness constraint*, *delay variation constraint*, and *non-overlapping spectrum constraint*. Due to space limitation, the definition of these constraints can be found in [6].

2.2 Notations

In this subsection, the notations used in the paper are defined. The physical topology of the network is denoted as $G = (V, E, delay)$, where $V = \{v_1, v_2,...,v_n\}$ is the set of nodes ($|V| = n$), $E = \{e_1, e_2, ..., e_m\}$ is the set of links ($|E| = m$), and $delay(e_l)$ is delay of the link $e_l \in E$. The request is defined as $r = (s, d, T_{sd}, \Delta_{sd})$, where $s \in V$ and $d \in V$ is the source and destination node of the request, respectively. T_{sd} is the required bandwidth (Gb/s) and Δ_{sd} is the maximum delay variation of the lightpaths between nodes s and d. $PS = \{(p_i, C_i, delay(p_i))|i = 1, 2, ..., |PS|\}$ is the set of current lightpaths used to support the original request $r = (s, d, T_{sd}, \Delta_{sd})$, where C_i is the number of FSs provided by the lightpath p_i, and $delay(p_i) = \sum_{\forall e_l \in p_i} delay(e_l)$ is the delay of the lightpath p_i. Let $MXD = \max\{delay(p_i), \forall p_i \in PS\}$, $MID = \min\{delay(p_i), \forall p_i \in PS\}$ and $MXD - MID \leq \Delta_{sd}$. $C^{total} = \sum_{\forall p_i \in PS} C_i$ is the total number of allocated FSs of the set PS of lightpaths, and $C^{total} \times C_f \geq T_{sd}$, where C_f is the bandwidth provided by a single FS. Let $r^{new} = (s, d, T^{new}_{sd}, \Delta^{new}_{sd})$ be the new request and B be the number of FSs provided with each link of the network.

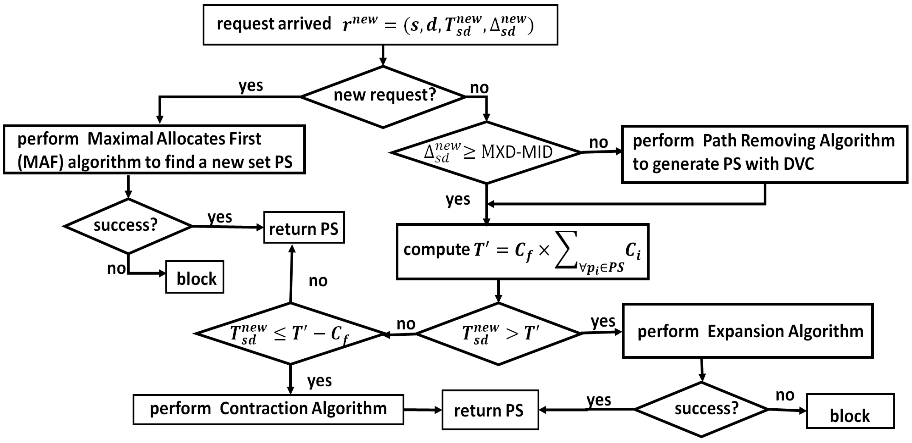

Fig. 2. Flowchart of SEC algorithm.

If the request can be supported by a single lightpath, the minimal required number of FSs can be computed by $\lceil T_{sd}/C_f \rceil$ plus a guard band (GB). That is, if $\lceil T_{sd}/C_f \rceil$ less than or equal to F, then the number of required GB is equal to 1. If $\lceil T_{sd}/C_f \rceil$ is greater than the upper bound F, or a block $\lceil T_{sd}/C_f \rceil$ of continuous FSs cannot be allocated on all links of the selected lightpath, the demand T_{sd} should be supported by establishing multiple lightpaths. In this case, more GBs will be introduced and the delay variation between established lightpaths should be satisfied. The minimal required number (N_{sd}) of FSs for the demand T_{sd} by using multipath routing can be computed by $N_{sd} = \lceil T_{sd}/C_f \rceil + \lceil \lceil T_{sd}/C_f \rceil / F \rceil \times GB$.

For a lightpath request $r = (s, d, T_{sd}, \Delta_{sd})$, the MAF algorithm proposed in [6] with online path computation is used to determine a set of routing paths PS=$\{p_i | i = 1, 2, ..., |PS|\}$ with DVC to serve the request. Note that, for different iteration, the routing paths can be identical, but since their spectrum allocations are not contiguous, more than one set of O-OFDM transceivers (TRs) are required.

3. PROPOSED ALGORITHMS

It is important to note that the value of DVC of an adjusted request may change, so that the currently routed lightpaths may not satisfy the DVC. In this situation, those lightpaths, which violate the DVC, should be deleted, and then, some new DVC-satisfied lightpaths may be added to fulfill the bandwidth requirement. In this section, proposed SEC algorithm which consists of several algorithms are proposed to do the SEC operation and described in the following subsections.

3.1 SEC Algorithm

The flowchart of the SEC Algorithm is shown in Fig. 2. When a request arrives, if it is a new request, then the MAF algorithm is performed to find a set of lightpaths to route the request. If it is an adjusted request, the delay-variation constraint of the currently deployed set PS of lightpaths should be checked in the initial step. For the new request $r^{new} = (s, d, T_{sd}^{new}, \Delta_{sd}^{new})$, if $\Delta_{s,d}^{new} \geq MXD - MID$ then all paths in PS satisfy the delay-variation constraint. If $\Delta_{s,d}^{new} < MXD - MID$, then some lightpaths in PS should be deleted to satisfy the DVC by performing the **Path Removing Algorithm** described in

Algorithm 2. After performing the Path Removing Algorithm, let $C^{total} = \sum_{\forall p_i \in PS} C_i$ be the total allocated FSs of the request and $T' = C^{total} \times C_f$ be the total bandwidth provided by the current set *PS* of paths.

It is worth noting that, the total bandwidth provided by the current set of lightpaths is greater or equal to the originally required bandwidth T_{sd}. Since the minimal unit of bandwidth allocation is a single FS and the smallest bandwidth provided with a slot is C_f. If $T' - C_f < T_{sd}^{new} \leq T'$ then there is no need to perform expansion or contraction. If $T_{sd}^{new} > T'$ then the **Expansion Algorithm** should be performed. If $T_{sd}^{new} \leq T' - C_f$ then the **Contraction Algorithm** should be performed.

In this **Expansion Algorithm**, the allocated FSs of current lightpaths are expanded first, if possible. If the required bandwidth cannot be fully supported, a set of new lightpaths is added to route the lack of bandwidth by performing the **Path Adding Algorithm (PAA)**. Two types of PAAs are proposed, they are PAA without deletion and PAA with deletion. (1) In **PAA without deletion**, all currently deployed lightpaths are not deleted in the path-adding process. (2) In **PAA with deletion**, if new lightpaths cannot be added due to the DVC, then some currently deployed lightpaths may be deleted and new lightpaths are found again to satisfy the constraint. If new lightpaths cannot be found to support the bandwidth, then the request is blocked.

When the bandwidth of the request decreases ($T_{sd}^{new} \leq T' - C_f$), the bandwidth of the currently allocated lightpaths may be decreased or lightpaths may be deleted by performing the **Contraction Algorithm**. The outline of the SEC Algorithm is described in Algorithm 1.

Algorithm 1 : SEC Algorithm

Input: $G(V,E,delay)$, $PS = \{(p_i, C_i, delay(p_i)) | i = 1,2,...,|PS|\}$, $r = (s,d,T_{sd},\Delta_{sd})$, $r^{new} = (s,d,T_{sd}^{new},\Delta_{sd}^{new})$, path-selecting policy (PSP);
Output: *PS*;
if ($r^{new} = (s,d,T_{sd}^{new},\Delta_{sd}^{new})$ is a new request) **then**
  Perform the MAF algorithm to find a set *PS* of lightpaths to route the request.
  If (*PS* can be found) **then return** *PS*; **else** block the request;
else
  compute *MXD* and *MID* of the *PS*;
  if ($\Delta_{sd}^{new} < MXD - MID$) **then**
    Perform **Path Removing Algorithm** to remove paths in *PS* which violates the DVC according to the PSP.
  Calculate the total bandwidth provided by the set *PS* of current lightpaths $T' = C_f \times \sum_{\forall p_i \in PS} C_i$.
  if ($T_{sd}^{new} > T'$) **then**
    Perform **Expansion Algorithm**.
    if (success) then return *PS*; **else** Block the request.
  else if ($T_{sd}^{new} \leq T' - C_f$) **then**
    Perform **Contraction Algorithm** to remove paths in *PS* according to the PSP and **return** *PS*.
  else
    return *PS*.
  end if
end if

3.2 Path Removing Algorithm

When the delay-variation constraint decreases and $\Delta_{sd}^{new} < MXD - MID$, currently allocated lightpaths should be updated by removing some lightpaths to satisfy the DVC. Therefore, the path-selection policies (PSPs) should be designed to select the currently allocated lightpaths for SEC operation (expansion, contraction or deletion). The priority or sequence of the selected paths can be determined by the PSPs. Two PSPs of Path Removing Algorithm are used. (1) Maximum delay first (MaxDF): the path with maximum delay is selected and removed. (2) Minimum delay first (MinDF): the path with minimum delay is selected and removed. For the case that there are several lightpaths with the same delay violate DVC, if selected the maximum or minimum delay, all lightpaths with the same delay should be removed. The details of the Path Removing Algorithm are described in Algorithm 2.

Algorithm 2 : Path Removing Algorithm

Input: $G(V,E,delay)$, $PS = \{p_i, C_i, delay(p_i) | i = 1,2,...,|PS|\}$, $r = (s,d,T_{sd}^{new},\Delta_{sd}^{new})$, $r = (s,d,T_{sd},\Delta_{sd})$, path-selecting policy (PSP), MXD, MID;
Output: PS;
while ($\Delta_{sd}^{new} < MXD - MID$) **do**
  Select a path p in PS with the highest priority according to the PSP.
  Remove all paths in PS which have the same delay as the selected lightpath p, restore network resources possessed by these lightpaths, and update MXD and MID.
end while
return PS.

3.3 Expansion Algorithm

When the bandwidth of the request increases, the bandwidth of currently allocated lightpaths should be increased or one (or several) new lightpath(s) with the required number of FSs should be added. First Let CP be the set of candidate paths and $CP = PS$. Based on the path-selecting policy, a lightpath p in CP is selected to expand. If the selected path p cannot be expanded further, then it was removed from CP. Otherwise, the maximal number C_p^e ($C_p + C_p^e \leq F$) of FSs can be expanded for the path p is found, where F is the maximum number of FSs supported by a lightpath. Let $C_r = \lceil (T_{sd}^{new} - T')/C_f \rceil$ be the required number of FSs of the path p and C_p be the number of current allocated FSs. The actual expanded number of FSs is determined by $C_e^x = \min\{C_p^e, C_r\}$.

If there is no path $p \in PS$ can be expanded, then the unsupported bandwidth will be supported by new lightpaths found by performing the **Path Adding Algorithm**, if possible. The details of the Expansion Algorithm are described in Algorithm 3. Four PSPs for Expansion Algorithm are used. (1) *Maximal Weight First* (MaxWF): the path in PS with maximal weight is selected first, the weight of the path is computed by the $(n - hop(p)) \times C_e^x$, where $hop(p)$ of is the hop of the path p. (2) *Minimal Weight First* (MinWF): the path in PS with minimal weight is selected first, the weight of the path is computed by the $hop(p) \times C_e^x$. (3) *Minimal Delay First* (MinDF): the path in PS with minimal delay is selected first. (4) *Maximal Delay First* (MaxDF): the path in PS with maximal delay is selected first.

3.4 Path Adding Algorithm

In the Path Adding Algorithm (PAA), some new lightpaths are added to support the lack of bandwidth of the request. Two types of PAA (without or with deletion) are proposed and described in this subsection.

3.4.1 PAA without deletion

In the *PAA without deletion*, all currently deployed paths in PS are kept. It is worth noting that the current paths in PS can be selected again and added to PS with different FS index since these paths satisfy the DVC.

If all new paths are selected from the set with a delay within $[MXD - \Delta_{sd}^{new}, MXD]$, they can be added to PS and satisfy the DVC. Moreover, the set of paths with a delay within $[MID, MID + \Delta_{sd}^{new}]$ has the same property. Therefore, a set P_{sd} of candidate paths with a delay within $[MXD - \Delta_{sd}^{new}, MID + \Delta_{sd}^{new}]$ are constructed. Then, the set P_{sd} is divided into two disjoint sets P_{sd}^1 and P_{sd}^2, where P_{sd}^1 contains paths in P_{sd} with a delay within $[MID, MXD]$ and $P_{sd}^2 = P_{sd} \setminus P_{sd}^1$. Paths in P_{sd}^1 can be added to PS without violating the DVC. Consider the example shown in Fig. 3, $\Delta_{sd} = 5$, the set of candidate paths P_{sd} is

Algorithm 3 : Expansion Algorithm

Input: $G(V,E,delay)$, $PS=\{(p_i,C_i,delay(p_i))|i=1,2,...,|PS|\}$, $r=(s,d,T_{sd}^{new},\Delta_{sd}^{new})$, path-selecting policy (PSP);
Output: PS^{new};
$MXD=\max_{\forall p_i\in PS}\{delay(p_i)\}$, $MID=\min_{\forall p_i\in PS}\{delay(p_i)\}$, $T'=C_f\times C^{total}$, $P^t=\emptyset$, $CP=PS$;
while $((T_{sd}^{new}>T')$ **and** $(CP\neq\emptyset))$ **do**
  Let $C_r=\lceil(T_{sd}^{new}-T')/C_f\rceil$ and select a path p in CP with highest priority according to the PSP.
  Find the maximal number C_p^e ($C_p^e+C_p\le F$) of expandable FSs for the selected path p.
  if $(C_p^e>0)$ **then**
    Temporarily allocate $C_e^x=\min\{C_p^e,C_r\}$ FSs to the path p. Update $T'=T'+C_e^x\times C_f$, path p and add path p to P^t.
  else
    Remove path p from CP.
  end if
end while
if $(T_{sd}^{new}>T')$ **then**
  Perform **Path Adding Algorithm** to find a set of new paths P^A to route the remaining required bandwidth.
  if (set P^A cannot be found) **then**
    Release all temporary allocation of lightpaths and **return false and** PS.
  else
    Allocate FSs to the paths in $P^t\cup P^A$, update PS by expanding paths in P^t and **return success and** $PS\cup P^A$.
  end if
else
  Allocate FSs to the paths in P^t, update PS by expanding paths in P^t and **return success and** PS.
end if

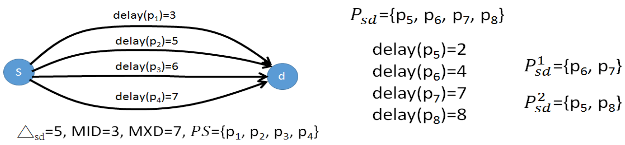

Fig. 3. Example of PAA.

$\{p_5, p_6, p_7, p_8\}$, $P_{sd}^1=\{p_6,p_7\}$ and $P_{sd}^2=\{p_5,p_8\}$. Paths in set $P_{sd}^1\cup PS$ can be selected as new paths and satisfy the DVC. First, the path p in $P_{sd}^1\cup PS$ is selected according to the PSP. Then, the maximal allocatable FSs C_p for the path p is found. If the path p can be allocated ($C_p>0$), then the maximal possible number of FSs ($C_p^e=\min\{F,C_p,C_r\}$) is allocated, where $C_r=\lceil(T_{sd}^{new}-T')/C_f\rceil$. If the required bandwidth cannot be satisfied, then the path-selecting process is repeated until no path in $P_{sd}^1\cup PS$ can be selected.

Each path p in P_{sd}^2 has the property that either $delay(p)>MXD$ or $delay(p)<MID$. If set P_{sd}^1 cannot be used to find new paths, the set P_{sd}^2 is used as the candidate set of paths. For the path $p\in P_{sd}^2$, we define $CD(p)=\min\{|delay(p)-MXD|,\ |delay(p)-MID|\}$, which is the delay that the difference $MXD-MID$ will increase, if the path p is selected and added to PS. The path p in P_{sd}^2 with minimal $CD(p)$ is selected and examined first.

Consider, the example shown in Fig. 4 (a), $\{p_1,p_2,p_3\}$ is the set P_{sd}^2 and these paths are listed increasingly according to the $CD(p)$. The path p_1 is the first selected path. After adding the path p_1 to PS, the MXD of PS should be updated, and if the path p_1 cannot be allocated again, then the set P_{sd}^2 is updated and sorted as $\{p_3,p_2\}$ (as shown in Fig. 4 (b)). That is, p_3 is the next selected path if needed. If the selected path p cannot be added to PS due to violating the DVC, then the request is blocked. Otherwise, the maximal allocatable FSs (C_p) for the path p is found and $C_p^x=\min\{F,C_p,C_r\}$ FSs are allocated. If the required bandwidth cannot be satisfied, then the path-selecting process is repeated until no path in P_{sd}^2 can be selected. The details of the Path Adding Algorithm (without deletion) are described in Algorithm 4.

3.4.2 PAA with deletion

In the *PAA with deletion*, currently deployed paths in PS may be deleted during the path adding process. A set P_{sd}, which includes the candidate paths from source s to

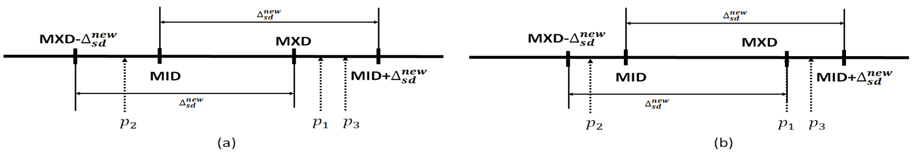

Fig. 4. Example of PAA without deletion for paths in P_{sd}^2.

Algorithm 4 : Path Adding Algorithm (without deletion)

1: **Input:** $G(V,E,delay)$, $PS=\{(p_i,C_i,delay(p_i))|i=1,2,...,|PS|\}$, $r=(s,d,T_{sd},\Delta_{sd})$; $r^{new}=(s,d,T_{sd}^{new},\Delta_{sd}^{new})$;
2: **Output:** PS^{new};
3: $MXD=\max_{\forall p_i\in PS}\{delay(p_i)\}$, $MID=\min_{\forall p_i\in PS}\{delay(p_i)\}$.
4: Find the set P_{sd} of lightpaths with a delay within $[MXD-\Delta_{sd}^{new}, MID+\Delta_{sd}^{new}]$. Divide the set P_{sd} into two disjoint sets P_{sd}^1 and P_{sd}^2, where P_{sd}^1 contains those paths in P_{sd} with a delay within $[MID,MXD]$ and $P_{sd}^2=P_{sd}\setminus P_{sd}^2$.
5: $PS^{new}=\emptyset$ and $CP=PS\bigcup P_{sd}^1$;
6: **while** $((T_{sd}^{new}>T')$ **and** $(CP\neq\emptyset))$ **do**
7: Select a path p in CP with highest priority according to the PSP.
8: Determine the maximal number of FSs (C_p) can be allocated, compute $C_r=\lceil(T_{sd}^{new}-T')/C_f\rceil$ and $C_p^x=\min\{F,C_p,C_r\}$.
9: **if** (there is free FSs for the selected path p i.e., $C_p^x>0$) **then**
10: Add the lightpath p to PS^{new} and PS and temporarily allocate C_p^x FSs to the path p, update T' and C_r.
11: **else**
12: Remove path p from CP.
13: **end if**
14: **end while**
15: **while** $((T_{sd}^{new}>T')$ **and** $(P_{sd}^2\neq\emptyset))$ **do**
16: For each path $p\in P_{sd}^2$, compute $CD(p)=\min\{|delay(p)-MXD|, |delay(p)-MID|\}$.
17: Sort paths in P_{sd}^2 in increasing order according to $CD(p)$ and select a path p in P_{sd}^2.
18: **if** (adding path p to PS^{new} can satisfy the delay-variation constraint) **then**
19: Determine the maximal number of FSs C_p can be allocated, $C_r=\lceil(T_{sd}^{new}-T')/C_f\rceil$ and $C_p^x=\min\{F,C_p,C_r\}$.
20: **if** $(C_p^x>0)$ **then**
21: Add p to PS^{new} and PS, update MXD, MID, and temporarily allocate C_p^x FSs to p, update T' and C_r.
22: **else**
23: Remove path p from P_{sd}^2.
24: **end if**
25: **else**
26: **break**;
27: **end if**
28: **end while**
29: **if** $(T_{sd}^{new}>T')$ **then** restore all resources of paths in PS^{new}, **return** false and $PS\setminus PS^{new}$; **else** allocate resources for paths in PS^{new}, **return** success and PS.

destination d, is found. Then, three subsets P_{sd}^A, P_{sd}^B and P_{sd}^C are constructed from P_{sd}. The path in P_{sd}^A is with a delay within [MIN, MXD]. The path in P_{sd}^B is with a delay less than MID, and the path in P_{sd}^C is with a delay greater than MXD. Consider the example shown in Fig. 3, if the value of Δ_{sd} is changed to 4 and the set of candidate paths is $\{p_5,$ $p_6,$ $p_7,$ $p_8\}$, then we have $P_{sd}^A=\{p_6,p_7\}$, $P_{sd}^B=\{p_5\}$ and $P_{sd}^C=\{p_8\}$. These sets are used as the candidate sets for adding new lightpaths. Let PS^{new} be the set of paths which have been selected as the new lightpaths for routing the demand of nodes s and d. The examining sequence of these sets are $P_{sd}^A\cup PS$, P_{sd}^B and P_{sd}^C. After considering the P_{sd}^C, if the requirement cannot be satisfied, then the request will be blocked. For the candidate set P_{sd}^A, the process is completely the same as the process in Algorithm 4 (steps 6–17) for the candidate set P_{sd}^1.

When the bandwidth requirement cannot be satisfied during considering the set P_{sd}^B, the maximum delay path p in P_{sd}^B is checked whether it can be added to PS, can satisfy the DVC (i.e., $MXD-delay(p)\leq\Delta_{sd}^{new}$) and can be allocated to the current network. If the path p can be allocated, the path p is added to the set PS^{new} and PS, and the demand C_r is updated by subtracting and allocating $C_p^x=\min\{F,C_p,C_r\}$ FSs for the path temporarily. Then, the current network G and bandwidth T' are updated accordingly. If $T_{sd}^{new}\leq T'$, then the set PS is returned; otherwise, the path adding process is repeated.

If path p cannot be added to PS due to the DVC cannot be satisfied, then those paths

p' in PS^{new} with $delay(p') > \Delta_{sd}^{new} - delay(p)$ are removed from PS and PS^{new}. Then, the network is updated by restoring the FSs allocated to those paths p' and the value of T' and C_r is restored by adding the demand allocated on the selected paths. Moreover, those paths whose delay are greater than $delay(p')$ are removed from the candidate set P_{sd}^{B}, since these paths cannot be selected as the lightpaths. The values of MXD of PS are also updated accordingly. If $T_{sd}^{new} \leq T'$, then the set PS is returned; otherwise, the path adding process is repeated.

The set P_{sd}^{C} is processed similarly, except for the order of the path is increasing order according to the delay of the path. The details of the Path Adding Algorithm (with deletion) are described in Algorithm 5.

Algorithm 5 : Path Adding Algorithm (with deletion)

Input: $G(V,E,delay)$, $PS = \{(p_i, C_i, delay(p_i)) | i = 1,2,...,|PS|\}$, $r = (s,d,T_{sd},\Delta_{sd})$, $r^{new} = (s,d,T_{sd}^{new},\Delta_{sd}^{new})$;
Output: PS^{new};
$MXD = \max_{\forall p_i \in PS}\{delay(p_i)\}$, $MID = \min_{\forall p_i \in PS}\{delay(p_i)\}$, find the set P_{sd}^{A} of paths with delay within $[MID, MXD]$.
$PS^{new} = \emptyset$, $CP = PS \bigcup P_{sd}^{A}$;
while $((T_{sd}^{new} > T')$ **and** $(CP \neq \emptyset))$ **do**
  Select a path p in CP with highest priority according to the PSP.
  Determine the maximal number of FSs C_p can be allocated, compute $C_p^x = \min\{F, C_p, C_r\}$ and $C_r = \lceil (T_{sd}^{new} - T')/C_f \rceil$;
  if $(C_p^x > 0)$ **then**
    Add the lightpath p to PS^{new} and PS and temporarily allocate C_p^x FSs to the path p and update T' and C_r.
  else
    Remove path p from CP.
  end if
end while
if $(T_{sd}^{new} \leq T')$ **then return** PS and allocate FSs to paths in PS^{new}.
Construct sets P_{sd}^{B} and P_{sd}^{C}, paths in P_{sd}^{B} and P_{sd}^{C} are sorted in decreasing order according to delay of paths.
while $((T_{sd}^{new} > T')$ **and** $(P_{sd}^{B} \neq \emptyset))$ **do**
  Select a path p from P_{sd}^{B} and find the maximal number C_p of allocatable FSs on G.
  if $(C_p == 0)$ **then**
    Remove path p from P_{sd}^{B}.
  else
    if (add path p to PS can satisfy DVC) **then**
      Compute $C_p^x = \min\{F, C_p, C_r\}$ and temporarily allocate C_p^x FSs on the path p.
      Add path p to PS^{new} and PS, update T', G, MXD, MID and C_r.
    else
      Release resources of $p' \in PS$ and PS^{new}, which violates DVC after adding path p $(delay(p') - delay(p) > \Delta_{sd}^{new})$.
      Restore T' from p', allocate C_p^x FSs to p, add path p to PS^{new} and PS, update T', G, MAD, MID and C_r.
    end if
  end if
end while
if $(T_{sd}^{new} \leq T')$ **then**
  return PS and allocate FSs to paths in PS^{new}.
end if
while $((T_{sd}^{new} > T')$ **and** $(P_{sd}^{C} \neq \emptyset))$ **do**
  Select a path p from P_{sd}^{C} and find the maximal number C_p of allocatable FSs on G.
  if $(C_p == 0)$ **then**
    Remove path p from P_{sd}^{C}.
  else
    if (add path p to PS can satisfy DVC) **then**
      Compute $C_p^x = \min\{F, C_p, C_r\}$ and temporarily allocate C_p^x FSs on the path p.
      Add path p to to PS^{new} and PS, update T', G, MXD, MID, C_r.
    else
      Release resources of $p' \in PS^{new}$ and PS, which violates DVC after adding path p $(delay(p) - delay(p') > \Delta_{sd}^{new})$.
      Restore the T' from path p' and temporarily allocate $C_p^x = \min\{F, C_p, C_r\}$ FSs on the path p.
      Add path p to to PS^{new} and PS, update T', G, MAD, MID and C_r.
    end if
  end if
end while
if $(T_{sd}^{new} > T')$ **then**
  Restore all temporarily allocated resources, and **return false** and $PS \setminus PS^{new}$.
else
  return PS and the allocated FSs to all paths in PS^{new}.
end if

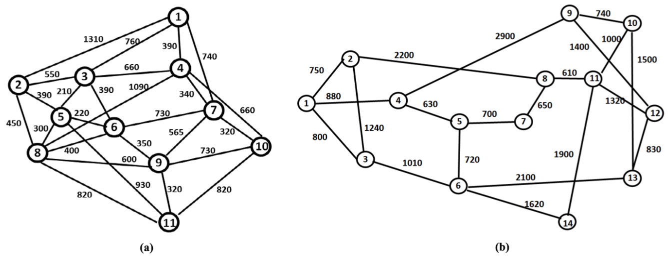

Fig. 5. (a) COST239 network; (b) NSFNET network.

3.5 Contraction Algorithm

When the bandwidth of the request decreases, the bandwidth of currently allocated lightpaths should be decreased or a (or several) lightpath(s) should be removed. In this case, the new DVC is greater than or equal to the original DVC; so that, after removing paths or contracting paths, the remaining paths still satisfy the DVC. The contraction process is repeated until the bandwidth requirement is satisfied.

Two possible actions can be applied for the contraction: lightpath deletion or lightpath contraction. For each option, there are many choices and the PSP should be designed to make the selection. Based on the PSP, a lightpath p in PS is selected to contract. Four PSPs for Contraction Algorithm are used: (1) *Maximal Weighted path First* (MaxWF): the path with maximal weight is selected first, the weight of the path p is defined as $FS(p) \times hop(p)$, where $FS(p)$ is the number of FSs allocated on the path p. (2) *Minimal Weighted path First* (MinWF): the path with minimal weight is selected first, the weight of the path p is defined as $FS(p) \times (n - hop(p))$, where $FS(p)$ is the number of FSs allocated on the path p. When two paths have the same number of allocated FSs, the path with greater hop is selected first. (3) *Minimal Delay First* (MinDF): the path in PS with minimal delay is selected first. (4) *Maximal Delay First* (MaxDF): the path in PS with maximal delay is selected first. The details of the Contraction Algorithm are described in Algorithm 6.

Algorithm 6 : Contraction Algorithm

Input: $G(V,E,delay)$, $PS = \{p_i, C_i, delay(p_i) | i = 1,2,...,z\}$, $r = (s,d,T_{sd}^{new},\Delta_{sd}^{new})$, path-selecting policy (PSP);
Output: PS;
Calculate the total bandwidth provided by the set of current lightpaths $T' = C_f \times \sum_{\forall p_i \in PS} C_i$.
if ($T_{sd}^{new} == 0$) **then**
    Remove all lightpaths in PS, restore network resources of lightpaths and **return** $PS = \emptyset$.
end if
while ($T_{sd}^{new} - T' \geq C_f$) **do**
    Select a path p in PS with highest priority according to the PSP.
    Find the number of allocated FSs C_p of the selected lightpath p and compute $C_c = \lfloor (T_{sd}^{new} - T')/C_f \rfloor$;
    if ($C_p \leq C_c$) **then**
        Remove lightpath p from PS and release resources of lightpath p and update $T' = T' - C_f \times C_p$;
    else
        Reduce the number of allocated FSs of p to $C_p - C_c$, update $T' = T' - C_f \times C_c$.
    end if
end while
return PS.

4. SIMULATION RESULTS

The proposed algorithms were coded by using C++ programming language. All simulations were run on a notebook computer with Intel Core i7-4710HQ CPU 2.5GHz, 16.0 GB RAM and with Windows 10 pro 64-bit operating system. Two topologies (COST239 and NSFNET showed in Fig. 5) were used for simulations, the number nears the link represents the delay of the link and the unit of it is a millisecond. The average delay of a pair of nodes on COST239 and NSFNET is 1.06 and 3.82 seconds, respectively. The initial traffic was randomly generated for all possible pairs of nodes and the maximal DVC of it is 2.8 seconds. The lightpaths of the initial traffic were established by performing the DVC-satisfy multipath routing MAF algorithm proposed in [6].

The adjusted traffic is randomly generated for all possible pairs of nodes with equal probability, the number of required FSs of the request is within [0, 40]. The arrival of an adjusted request to the network follows the Poisson distribution with a mean value of λ requests per unit time, the connection-holding time obeys the negative exponential distribution with a mean value of $1/\mu$ and 4000 adjusted requests are randomly generated. These requests are simulated for each algorithm for several different network loads. The network load is given as λ/μ Erlangs (λ/μ in $\{400, 800, 1200, 1400, 1600, 1800, 2000, 2200, 2400\}$). For each fiber, 100 and 120 FSs are provided by the COST239 and NSFNET network, respectively. The maximal DVC of adjusted requests is in $\{5, 6, 7, 8, 9\}$. Several performance criteria are considered in this paper, they are: (1) Blocking Ratio (BR): BR is defined as the ratio of blocked adjusted requests versus the total number of adjusted requests; (2) the average number of TRs of request: this is the average number of used TRs (which is propositional to the number of used GBs) for all active requests; and (3) computation time.

4.1 Performance of PSP

In this subsection, the performance of the PSP is examined. The PSPs proposed in Section 3 for the path removing, expansion and contraction algorithms are compared through simulations. The simulation network is the COST239 network. For these PSPs, the simulation results for different network loads (in Erlang) of adjusted requests are shown in Fig. 6. In these simulations, the maximal number of FSs (F) can be transmitted by a TR is set to 4 and the maximal DVC of the adjusted request is set to 6 seconds.

In most of the cases, the computation time for these PSPs has no significant difference, only the results for BR and the number of used TRs are shown here. For the simulation of the path removing algorithm (shown in Figs. 6 (a) and (d)), the PSPs of expansion and contraction algorithms are not changed and set as the MinDF. In Fig. 6 (a), the MaxDF scheme can get a lower BR than that of the MinDF. This may be the reason that keeps the delay of paths in current set *PS* lower may have a better chance to find routing paths and network resources for the adjusted request. In Fig. 6 (d), the MaxDF scheme can use a less number of TRs than that of the MinDF.

For the simulation of the expansion algorithm (shown in Figs. 6 (b) and (e)), the PSPs of path removing and contraction algorithms are not changed and set as the MaxDF and MinDF, respectively. In Fig. 6 (b), the MaxDF scheme can get a lower BR than that of the other methods. This may be the reason that adding the path with greater delay of paths in the candidate set to the *PS* may have a better chance to find routing paths and network resources for the expanded request. In Fig. 6 (e), the MinDF scheme can use a less number of TRs than that of the other methods.

For the simulation of the contraction algorithm (shown in Fig. 6 (c)), the MaxWF

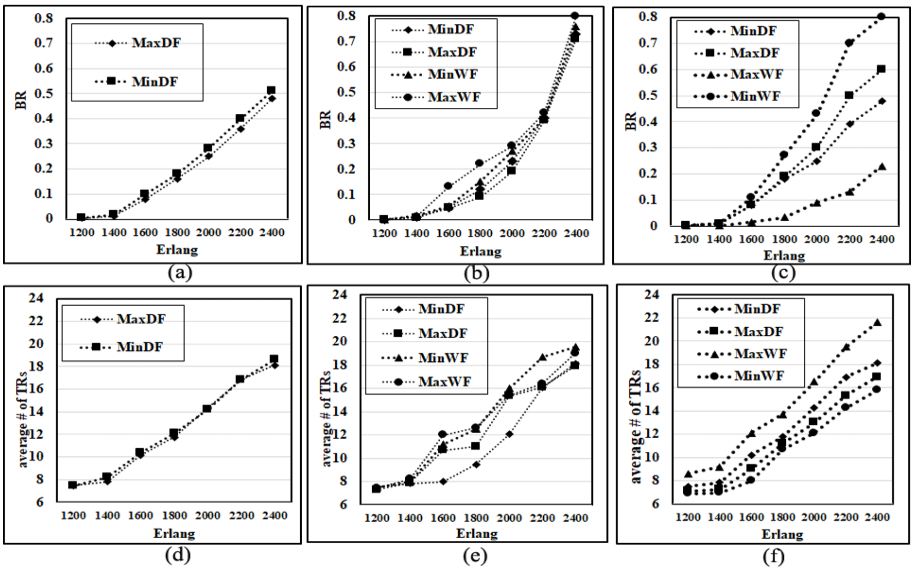

Fig. 6. Simulation results for the PSPs on COST239 network of the (a)(d) path removing algorithm, (b)(e) expansion algorithm, (c)(f) contraction algorithm.

method can get a lower BR on the network than that of the other methods. This may be the reason that releasing (or contracting) the path with greater weight (computed by $hop(p)\times$ the number of allocated FSs) in PS may release resources holding by the path with greater hops, so that it can reduce the resource usage and has a better chance to route subsequent requests successfully. The MinWF can use a less average number of TRs than that of the other methods (shown in Fig. 6 (f)). Since the path with small allocated FSs can be released so that the number of used TRs and GBs can be reduced.

4.2 Comparisons

To know the performance of the SEC operation, an algorithm denoted as 'release-and-add' was implemented for comparison. In the release-and-add method, for each adjusted request, first, the currently established lightpaths and the allocated resources for the original request are released; and then a set of new lightpaths are re-established by performing the MAF algorithm if needed.

For the COST239 network with 100 FSs, for different network load, the simulation results are shown in Fig. 7. Fig. 7 (a) shows that the PAA with deletion method can get a lower BR than that of the PAA without deletion and release-and-add methods. Since the PAA with deletion method can search more candidate paths to meet the DVC but with the cost of computation time. The value of BR increases as the network load increases. For the average number of used TRs, Fig. 7 (b) shows that the release-and-add uses more TRs than other methods. Since the release-and-add method may establish more lightpaths with less number of FSs to meet the bandwidth requirement, result in using more TRs and GBs. Fig. 7 (c) shows that the release-and-add method is the quickest method. This may be the reason that the release-and-add method simply released all currently deployed lightpaths and find a set of new lightpaths, it can be performed quickly. Fig. 7 (c) also shows that the PAA without deletion is faster than the PAA with deletion. This may be the reason that in the PAA with deletion, more lightpaths and processes are considered than the PAA without deletion. The CPU time also increases as the network load increases for these

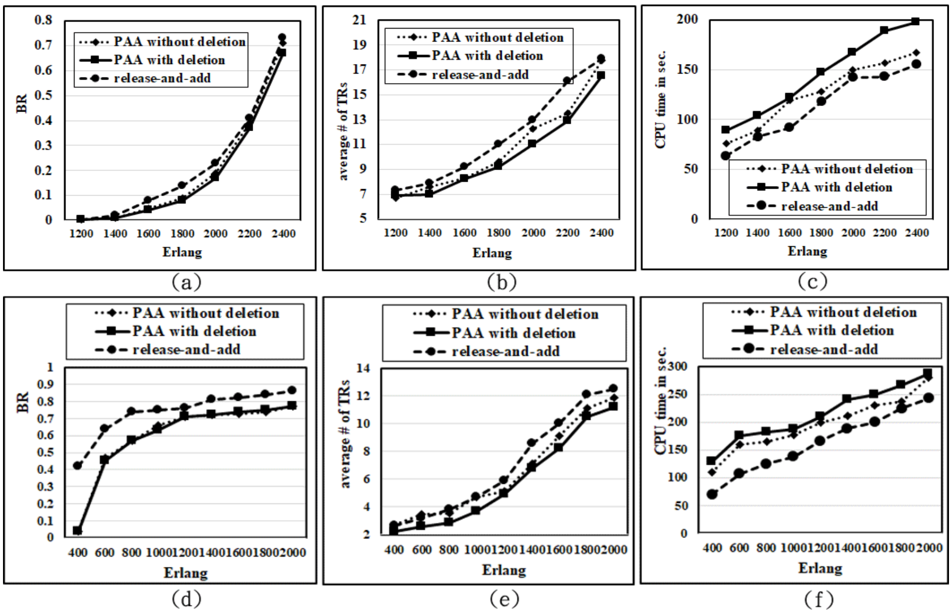

Fig. 7. Simulation results on COST239 network with 100 FSs (a) BR; (b) Average number of TRs; (c) CPU time in seconds, simulation results on NSFNET network; (d) BR; (e) Average number of TRs; (f) CPU time in seconds.

algorithms. The similar results for the NSFNET network with 120 FSs can be found in Figs. 7 (d)-(f).

For the network at a load of 2000 Erlangs, different maximal DVCs in {5, 6, 7, 8, 9} of adjusted requests, the simulation results on the COST239 network with 100 FSs per link are shown in Figs. 8 (a)-(c). For the network at a load of 800 Erlangs, different maximal DVCs in {5, 6, 7, 8, 9} of adjusted requests, the simulation results for NSFNET network with 120 FSs per link are shown in Figs. 8 (d)-(f). Figs. 8 (a) and (d) show that the PAA with deletion can get a lower BR than that of the PAA without deletion and release-and-add methods. In the path-adding, the PAA with deletion is more flexible than the PAA without deletion and can use spectrum more efficient than the release-and-add method. The results show that as the value of DVC increases, the value of BR decreases since it is easier to find constraint-satisfied routing lightpaths. Moreover, the COST239 network is denser than the NSFNET network, for the same network load, the BR on COST239 network is less than that on the NSFNET network. The difference of BR of PAAs and release-and-add is about 30% on the COST239 network and it is higher than that of the NSFNET network (about 15%). For the average number of used TRs, Figs. 8 (b) and (d) show that the PAA with deletion uses a less number of TRs than other methods. Figs. 8 (c) and (f) show that the release-and-add method is the fastest one and the PAA with deletion is the slowest one.

For the network at a load of 2000 Erlangs, maximal DVC 6, for different values of F in {4, 5, 6, 7, 8, 9, 10, 11, 100}, the simulation results for COST239 with 100 FSs per link are shown in Figs. 9 (a)-(c). For the network at a load of 800 Erlangs, the simulation results for NSFNET with 120 FSs per link are shown in Figs. 9 (d)-(f). The F is set to 100 for the COST239 network (or 120 for the NSFNET network) means that there is no restriction on the maximal number of FSs for a transceiver (or a single lightpath). In

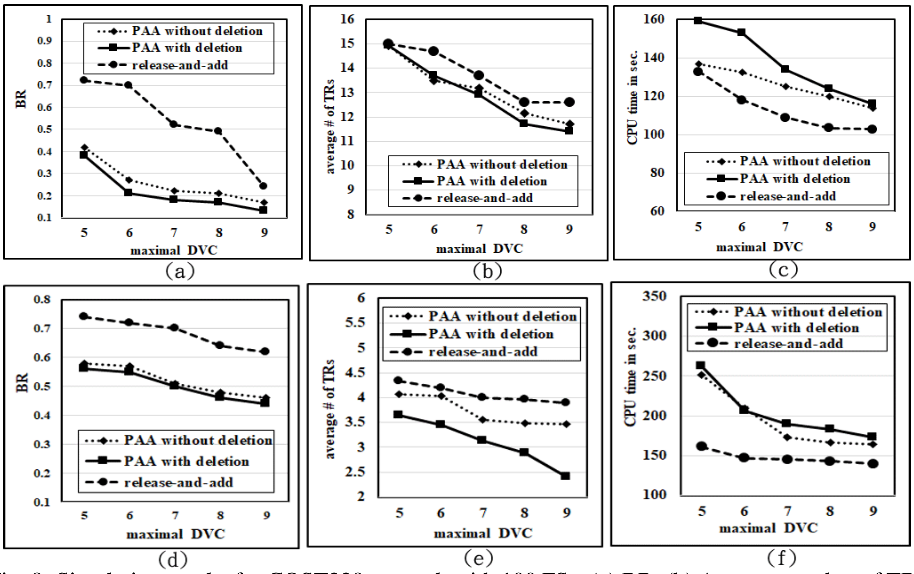

Fig. 8. Simulation results for COST239 network with 100 FSs; (a) BR; (b) Average number of TRs; (c) CPU time in seconds, simulation results for NSFNET network; (d) BR; (e) Average number of TRs; (f) CPU time in seconds.

this case, if there are available FSs for the request, the request will be routed by a single lightpath and the numbers of GBs and used TRs can be reduced. The adjusted request is blocked may due to delay-variation constraint or lack of network resources. As the value of F increases, the value of BR decreases in both networks (shown in Figs. 9 (a) and (d)). This may be the reason that lightpaths with more FSs can be found to route the request so that the number of GBs can be reduced. Figs. 9 (a) and (d) show that the PAA with deletion can get a lower BR than that of the other methods. For the average number of used TRs, Figs. 9 (b) and (e) show that the PAA with deletion uses a less number of TRs than that of the other methods. The release-and-add method is the worst one. Figs. 9 (c) and (f) show that the release-and-add method is the fastest one and the PAA with deletion is the slowest one.

5. CONCLUSIONS

In this paper, the delay-variation constrained spectrum expansion and contraction problem for multipath routing with time-varying traffic on EONs has been studied. For a given EON and a set of requests, the goal is to design a spectrum expansion and contraction method to update the CF and the channel size of the lightpath to fit the required of the request. In the studied problem, the multiple-path routing scheme is used.

In this paper, several heuristic algorithms (Path-Removing, Expansion, Path Adding and Contraction) were proposed to perform the traffic update. To select lightpaths for updating in multipath routing scheme, several path-selecting policies (PSPs) are designed for each algorithm. Two path adding algorithms (PAAs) are proposed to find DVC-satisfy routing lightpaths. The simulations show that the PAA with deletion can get the lowest BR than other methods. Compare to the release-and-add method, it can reduce BR about 7%.

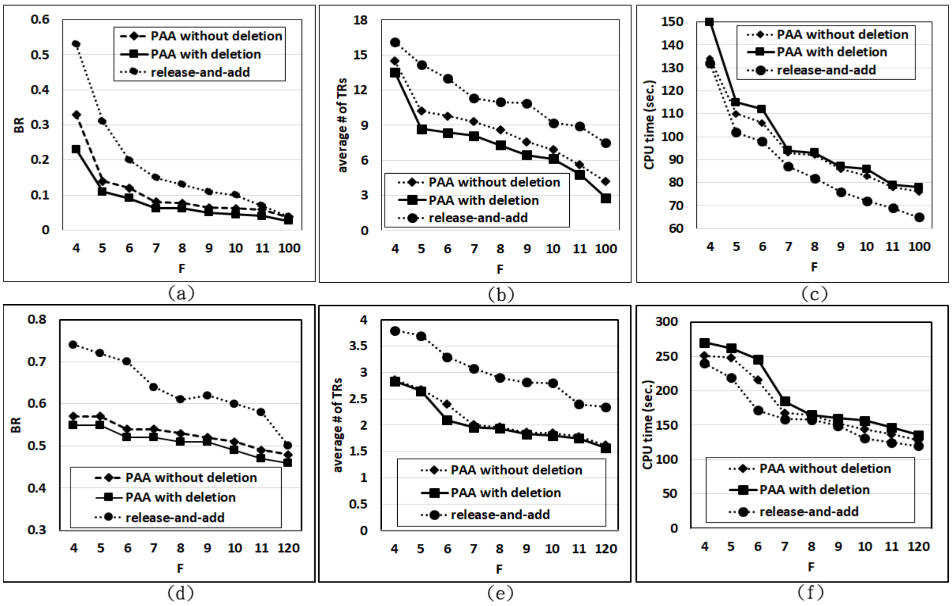

Fig. 9. Simulation results for COST239 network with 100 FSs; (a) BR; (b) Average number of TRs; (c) CPU time in seconds, simulation results for NSFNET network; (d) BR; (e) Average number of TRs; (f) CPU time in seconds.

REFERENCES

1. G. Zhang, M. D. Leenheer, A. Morea, and B. Mukherjee, "A survey on OFDM-based elastic core optical networking," *IEEE Communications Surveys & Tutorials*, Vol. 15, 2012, pp. 65-87.
2. K. Klinkowski and L. Walkowiak, "Routing and spectrum assignment in spectrum sliced elastic optical path network," *IEEE Communications Letters*, Vol. 15, 2011, pp. 884-886.
3. A. Shami, C. Assi, I. Habib, and M. A. Ali, "Performance evaluation of two GMPLS-based distributed control and management protocols for dynamic lightpath provisioning in future IP networks," in *Proceedings of International Conference on Communications*, 2002, pp. 2289-2293.
4. Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *Journal of Lightwave Technology*, Vol. 31, 2013, pp. 15-22.
5. W. Lu, X. Zhou, L. Gong, M. Zhang, and Z. Zhu,"Dynamic multi-path service provisioning under differential delay constraint in elastic optical networks," *IEEE Communications Letters*, Vol. 15, 2013, pp. 158-161.
6. D. R. Din, "Heuristic and genetic algorithms for solving the virtual topology design problem on elastic optical networks," *Journal of Information Science and Engineering*, Vol. 33, 2017, pp. 305-327.
7. G. Shen, Q. Yang, S. You, and W. Shao, "Maximizing time-dependent spectrum sharing between neighbouring channels in CO-OFDM optical networks," in *Proceedings of International Conference on Transparent Optical Networks*, 2011, pp. 1-4.

8. M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," *IEEE Journal on Selected Areas in Communications*, Vol. 31, 2013, pp. 26-38.
9. K. Christodoulopoulos, I. Tomkos, and E. Varvarigos "Time-varying spectrum allocation policies and blocking analysis in flexible optical networks," *IEEE Journal on Selected Areas in Communications*, Vol. 31, 2013, pp. 1-13.
10. C. Bijoy, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Communications Surveys & Tutorials*, Vol. 17, 2015, pp. 1776-1800.
11. A. Asensio, K. Klinkowski, M. Ruiz, V. Lopez, A. Castro, L. Velasco, and J. Comellas, "Impact of aggregation level on the performance of dynamic lightpath adaptation under time-varying traffic," in *Proceedings of the 17th International Conference on Optical Network Design and Modeling*, 2013, pp. 184-189.
12. I. Olszewski, "Dynamic RSA problem for time-varying traffic in spectrum sliced elastic optical path network," *International Journal of Electronics and Telecommunications*, Vol. 61, 2015, pp. 179-184.
13. P. S. Khodashenas, J. Comellas, S. Spadaro, J. Perello, and G. Junyent, "Using spectrum fragmentation to better allocate time-varying connections in elastic optical networks," *Journal of Optical Communications and Networking*, Vol. 6, 2014, pp. 433-440.
14. M. Chino, M. Kamio, J. Matsumoto, E. Oki, S. Okamoto, and N. Yamanaka, "Adaptive elastic spectrum allocation based on traffic fluctuation estimate under time-varying traffic in flexible OFDM-based optical networks," *IEICE Transactions on Communications*, Vol. E100.B, 2017, pp. 962-973.
15. D. R. Din, Y. F. Wu, B. J. Guo, C. Chen, and P. J. Wu, "Spectrum expansion/contraction problem for multipath routing with time-varying traffic on elastic optical networks," in *Proceedings of International Conference on Internet of Things, Data and Cloud Computing*, 2017, Article No. 51.
16. S. Zhang, *et al.*, "Dynamic traffic grooming in elastic optical networks," *IEEE Journal on Selected Areas in Communications*, Vol. 31, 2013, pp. 4-12.

Der-Rong Din (丁德榮) received M.S. and Ph.D. degrees in Computer and Information Science from National Chiao-Tung University in 1993 and 2001, respectively. Now, he is a Professor in the school of Department of Computer Science and Information Engineering at National Changhua University of Education. Professor Din has published over fifty articles on the theory and practice of optimization-based problem. His research interests include routing, multicast routing, survivability, network design, network planning in optical networks.