JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 159-174 (2019)
DOI: 10.6688/JISE.201901_35(1).0009

Survivable Routing Problem in EONs with FIPP *p*-Cycles Protection*

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

In this paper, the survivable routing problem on *elastic optical networks* (EONs) for the single link-failure case is studied. The *failure-independent path-protecting p-cycles* (FIPP *p*-cycles) protection scheme is used for lightpath protection. For the given network and the set of connection requests, the lightpaths used to route the connection requests and the protecting cycles are found. In this paper, two new properties of FIPP *p*-cycles are introduced on EONs to improve the spectrum efficiency. Several heuristic algorithms are proposed to solve this problem and simulations are run in the static case to minimize the total number of frequency slots. The performance of fragmentation ratio and resource utilization ratio are also examined.

Keywords: elastic optical network, heuristic algorithm, FIPP *p*-cycles, survivable routing, fragmentation

1. INTRODUCTION

Elastic optical networks (EONs) possess the benefits of high spectrum efficiency and flexible bandwidth allocation [1]. The spectrum of a fiber in EONs is divided into a small unit (called *frequency slots* (FSs)), and a necessary amount of consecutive FSs for a given data rate is assigned to support the connection request. Moreover, EONs provide a super-channel connectivity for accommodating ultra-high capacity demands and a sub-wavelength granularity for low-rate transmissions [1]. In [2], authors used the *layered graph RSA* (LG-RSA) to design integrated *multicast-capable routing and spectrum assignment* algorithm for achieving efficient all-optical multicast on EONs. This approach can also be used to solve the RSA problem for unicast on EON networks and can find routing path and FSs simultaneously.

A network failure (such as a fiber cut) may cause tremendous data loss. When the failed link is located in a lightpath of a connection, the traffic will be affected. *Survivability* is an important issue in EONs [3]. Many *protection* and *restoration* schemes have been proposed to achieve survivability in the optical layer [3] on EONs. Protection is a proactive scheme, some resource reservations are done for a connection before it fails. When setting up optical paths on networks, it is often desirable to set up a link-disjoint protection path at the same time.

In [4], the *failure-independent path-protecting p-cycles* (FIPP *p*-cycles) was proposed for *wavelength-division multiplexing* (WDM) network to protect connections. The FIPP *p*-cycles scheme is known to be able to achieve both fast restoration speed and high

Received May 22, 2017; revised July 22, 2017; accepted September 24, 2017.
Communicated by Xiaohong Jiang.
* This work was supported in part by the MOST project under Grant 105-2221-E-018-018.

spectrum efficiency [4]. More details about the FIPP p-cycles scheme can be found in [4, 5]. In [6, 7], the FIPP p-cycles scheme is extended and applied on EONs. Although several types of research for survivable routing with FIPP p-cycles on EONs have been studied [6-10], there is no study related to the expandable FS allocation scheme has been proposed so far. Moreover, the new feature that releases the link-disjoint constraint of the FIPP p-cycles has not been examined in EONs.

In this article, the survival routing problem with FIPP p-cycles protecting scheme is studied on EONs and the aim is to provide 100% restorability for lightpaths during any single link-failure. An expandable FS allocation scheme in EONs is considered to increase the efficiency of the backup resources. Then, several heuristic algorithms are proposed to solve this problem and evaluated by extensive simulations.

This rest of the article is organized as follows. In section 2, the related works are described. In section 3, the motivation and contribution of this paper are described. In section 4, the definition of the problem is described. The proposed heuristic algorithms are described in section 5. Experimental results and conclusions are given in sections 6 and 7, respectively.

2. RELATED WORKS

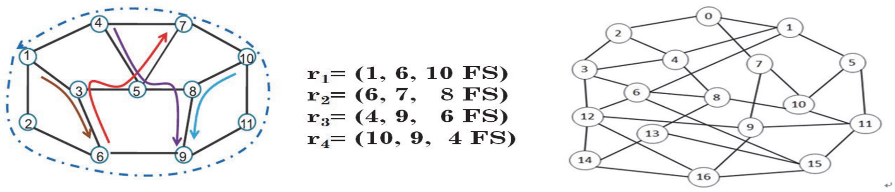
2.1 FIPP *p*-Cycles

In order to protect working paths in WDM networks, Kodian and Grover [4] have proposed the FIPP p-cycles protecting scheme. Protection designs with FIPP p-cycles have been intensively studied for WDM networks [4, 5]. In the FIPP p-cycles protecting scheme, several mutually link-disjoint lightpaths form a disjoint route set (DRS) [4]. For each DRS, an FIPP p-cycle, which passes through all end-nodes of the lightpaths in the DRS, is established to protect the set of lightpaths. In WDM, the bandwidth of each request is one wavelength, thus the FIPP p-cycle also needs one wavelength. Since lightpaths in the DRS are link-disjoint, only one lightpath will be affected by a single-link failure. More demands can be protected by a single FIPP p-cycle, which yields a better resource redundancy. For a given protecting cycle, it can provide two protecting paths for the straddle lightpath and a protecting path for the on-cycle lightpath [4].

2.2 FIPP *p*-Cycles on EONs

The FIPP p-cycles protecting scheme has been studied for protecting EONs [6-10]. The implementation of FIPP p-cycles on EONs was first considered in [6], where the authors designed a heuristic algorithm to solve the survival routing and spectrum assignment (SRSA) problem. However, they did not address the spectrum-efficient FIPP p-cycles design together with RSA and the FIPP p-cycle is found simply by combining two link-disjoint lightpaths in EONs.

In [7], authors studied the problem of offline service provisioning with FIPP p-cycles, they designed an integer linear programming (ILP) model and proved the problem is NP-hard. Moreover, authors [7] also considered an online service provisioning with FIPP p-cycles on EONs. Several heuristic algorithms were designed for finding routing paths and FIPP p-cycles protection. They also proposed a p-cycle reconfiguration scheme to re-optimize protection structures dynamically. Consider the example shown in Fig. 1 (a) [7], the routing paths and the protecting cycle of the four connection requests are given. In [7], authors stated that the working lightpaths 10→8→9 and 4→5→8→9 pass through the same link (8, 9), thus at most three lightpaths of these lightpaths can be protected by the selected cycle due to the link-disjoint constraint of FIPP *p*-cycles scheme. Thus, the Maximum-Independent-Set (MIS)-FIPP algorithm was proposed to find the maximum independent set of lightpaths [7].



(a) (b)

Fig. 1. (a) Example of FIPP *p*-cycle protecting scheme used in [7]; (b) simulation network.

In [8, 9], the FIPP *p*-cycles protecting scheme were used for *space division multiplexing* EONs (SDM-EONs). In [8], the FIPP *p*-cycles for SDM-EONs without the minimum interference criteria was considered. In [9], authors introduced a novel algorithm to provide FIPP *p*-cycles with minimum interference for path protection in SDM-EONs. The proposed algorithm prioritizes the use of straddling *p*-cycles in order to generate minimum interference to reduce rejections of future requests. Results in [9] showed that the proposed algorithm promotes protection effectively without compromising networking blocking. In [10] incorporated bandwidth-squeezed restoration to design two novel availability-aware service provisioning (AaSP) schemes with and without topology partition. Their simulation results showed that compared with the existing approach proposed in [7], the proposed algorithms achieved 50% spectrum-saving.

3. MOTIVATION AND CONTRIBUTION

In this section, the motivation and the major contribution of proposed methods are described. To improve the spectrum efficiency of the FIPP *p*-cycles protecting scheme, two modifications for allocating FIPP *p*-cycles are introduced in this paper. They are *p*-cycle with expandable FS allocation and releasing of link-disjoint constraint.

3.1 Expandable FS Allocation

For a set of working paths $P = \{P_1, P_2, ..., P_k\}$, let $FS(P_i)$ be the number of required FSs of the path $P_i \in P$. If all paths in P are link-disjoint and a directional cycle, which passes through all end-nodes of paths, is used to protect these working paths, then $FS(P) = \max\{FS(P_1), FS(P_2), ..., FS(P_k)\}$ FSs should be allocated on the cycle. Consider the example shown in Fig. 2 (a), P_1, P_2 and P_3 are link-disjoint lightpaths and a clockwise directional cycle is used to protect these lightpaths, the number of required FSs of the cycle is $\max\{FS(P_1), FS(P_2), FS(P_3)\}$.

In the online service environment, the connection requests are added or deleted dynamically. In the dedicated protecting scheme, terminate a connection is very simple, *i.e.*, the working and backup lightpaths of the connection can be released directly without affecting the survivability of other connections. However, in the FIPP p-cycles protecting scheme (or other shared backup paths protecting schemes), the shared backup resources of the protecting cycle cannot be released immediately, if there still exists any lightpath protected by the cycle. In the dynamic environment on EONs, the required backup resources of the protecting cycle may be decreased/expanded/deleted/unchanged accordingly. Consider the example shown in Fig. 2 (a), if the path P_3 is deleted, then the number of allocated FSs for the protecting cycle can be reduced from $\max\{FS(P_1), FS(P_2), FS(P_3)\}$ to $\max\{FS(P_1), FS(P_2)\}$. If $\max\{FS(P_1), FS(P_2)\} < \max\{FS(P_1), FS(P_2), FS(P_3)\}$, then the number of required FSs can be reduced.

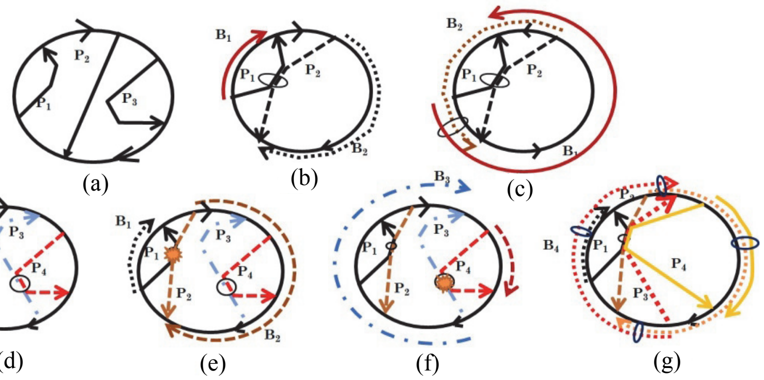


Fig. 2. Example of the problem; (a) link-disjoint lightpaths; (b) overlapped lightpaths; (c) protecting paths for (b) if common link failed; (d) more complex example; (e) protecting path for (d); (f) protecting path for (d); (g) more complex example with overlapped lightpaths.

For a new connection request, if the lightpath for the request is link-disjoint to all lightpaths protected by the selected cycle, then the lightpath can be added to the DRS of the cycle and then the number of allocated FSs of the cycle is expanded if possible. Consider the example shown in Fig. 2 (a), assume two lightpaths P_1 and P_2 are protected by the cycle and the new lightpath P_3 is added to the DRS. If $FS(P_3) \leq \max\{FS(P_1), FS(P_2)\}$, then the cycle can be used to protect these three lightpaths without FS expansion. For the case that $\max\{FS(P_1), FS(P_2)\} < FS(P_3)$, the cycle can be used to protect these three lightpaths after expanding $FS(P_3) - \max\{FS(P_1), FS(P_2)\}$ FSs, if possible. Otherwise, another or a whole new cycle should be found to protect the working path.

3.2 Releasing of Link-Disjoint Constraint

For the FIPP p-cycles protecting scheme, if the link-disjoint constraint for all working paths in a DRS is released and replaced by allowing their protecting paths are not disrupted, then the resource sharing may increase. This scheme is denoted as protection compatible. Consider the case that two working paths P_1 and P_2 are overlapped and a clockwise directional cycle is used to protect these lightpaths, the protecting paths B_1 and

B_2 are link-disjoint (the example shown in Fig. 2 (b)). Thus, the required number of FSs is $\max\{FS(P_1), FS(P_2)\}$. If the failed link is one of the common links, the protecting cycle can be used to restore these connections.

For the same set of lightpaths P_1 and P_2, but a counter-clockwise directional cycle is used to protect these lightpaths. Their protecting lightpaths are overlapped (shown in Fig. 2 (c)), if this cycle is used to protect these lightpaths, the required number of FSs is $FS(P_1) + FS(P_2)$. This example shows that the direction of the protecting cycle does affect the required number of FSs.

A more complex case that protecting paths are allowed jointly is considered in Fig. 2 (d). The sets $\{P_1, P_2\}$ and $\{P_3, P_4\}$ are protection compatible sets, but sets $\{P_1, P_3\}$, $\{P_1, P_4\}$, $\{P_2, P_3\}$ and $\{P_2, P_4\}$ are link-disjoint paths sets. To protect the lightpaths in set $\{P_1, P_3\}$, the selected cycle required $FS(P_1) + FS(P_3)$ FSs. For the other sets, the computation method is the same. The protecting paths for all lightpaths in Fig. 2 (d) after common link fail are shown in Figs. 2 (e) and (f). To use a single cycle to protect all lightpaths, the number of required FSs of the selected cycle can be computed by $\max\{\max\{FS(P_1), FS(P_2)\}, \max\{FS(P_3), FS(P_4)\}, \max\{FS(P_1), FS(P_2), FS(P_3), FS(P_4)\} = \max\{\max\{FS(P_1), FS(P_2)\}, \max\{FS(P_3), FS(P_4)\}\}$.

Similar case shown in Fig. 2 (f), the number of required FSs of the selected cycle can be computed by $\max\{FS(P_1) + FS(P_3), FS(P_2) + FS(P_3), FS(P_2) + FS(P_4), FS(P_1), FS(P_2), FS(P_3), FS(P_4)\} = \max\{FS(P_1) + FS(P_3), FS(P_2) + FS(P_3), FS(P_2) + FS(P_4)\}$. In general, for a set of lightpaths $P = \{P_1, P_2, ..., P_k\}$, let $PS(e_l)(PS(e_l) \subset P)$ be the set of lightpaths, which passes through link e_l and $FS(PS(e_l))$ be the least number of FSs for the cycle to protect paths in $PS(e_l)$. A FIPP p-cycle, which can be used to protect the set P of paths, should be allocated at least $FS(P) = \max_{\forall e_l \in E}\{FS(PS(e_l))\}$. Consider the example shown in Fig. 1, if the link-disjoint constraint were released, to protect these requests, 10 FSs should be allocated to the cycle. If the link (8, 9) fails, four lightpaths can be recovered by the selected cycle. These special features for the FIPP p-cycles on the EON was not considered in the previous studies and it motivates us to re-studied and re-examined its performance on EONs.

4. PROBLEM DEFINITION

4.1 Notations

This subsection gives notations using in this paper. Let $G(V, E)$ be the physical topology of a given EON, where $V = \{v_i \mid i = 1, 2, ..., |V|\}$ and $E = \{e_{ij} = (v_i, v_j) \mid v_i, v_j \in V\}$ is the set of nodes and fibers, respectively. Let e_{ij} be the fiber connecting nodes v_i and v_j. The connection request is represented by $r_i = (s_i, d_i, f_i)$, where $s_i \in V$ is the source node, $d_i \in V$ the destination node, and f_i is the number of required FSs. Let p_i be the lightpath for the connection request r_i. $P = \{P_{s_i,d_i} \mid \forall s_i, d_i \in V\}$ is the set of candidate paths for the connection (s_i, d_i, f_i). Let $C = \{c_1, c_2, ..., c_{|C|}\}$ be the set of all candidate FIPP p-cycles on the network G. C_{s_i,d_i} is the set of all candidate cycles which pass through end-nodes s_i and d_i in C ($C_{s_i,d_i} \subseteq C$) and $C^c \subseteq C$ is the set of FIPP p-cycles which are used to protect current active lightpaths. Let B be the number of frequency slots of each link on network G. For each fiber, B frequency slots are provided. Let B_{lj} be the number of occupied frequency slots on link $e_l \in E$ between jth and $(j + f_i - 1)$th FSs. Let $b_l(j)$ be the status of the link e_l

$\in E$ on jth FS which is a binary valuable indicator; $b_l(j) = 0$ means that the jth frequency slot of link e_l is free; otherwise, $b_l(j) = 1$.

4.2 Objective Function

In the paper, the dynamic environment is considered. For a given connection request (s_i, d_i, f_i), a lightpath is found as the working path to transmit the traffic demand and a protecting FIPP *p*-cycle is found and allocated to provide 100% protection of the request. The goal is to minimize the total allocated working and protecting FSs for all requests.

4.3 Assumptions

The assumptions of the problem on EONs for the single link-failure case are given as follows; (1) The physical network is a two-connected network; (2) For each link in the physical network, there is a fiber connecting the end-nodes; (3) All nodes in the network are with bandwidth-variable transponders (BVT) and bandwidth-variable cross-connects (BV-WXC); (4) For simplicity, the numbers of FSs provided by links are all equal; (5) At each instance, only a single link may fail.

4.4 Constraints

On EONs, several constraints should be satisfied, they are *spectrum continuity constraint*, *subcarrier consecutiveness constraint*, *and non-overlapping spectrum assignment constraint*. Due to the limitation of the article, the definitions of the constraints can be found in [4, 7] and will not be stated here.

5. PROPOSED ALGORITHMS

In this section, the details of the proposed algorithms are described. The problem is solved by dividing into two (*path finding and cycle finding*) subproblems. In path finding and cycle finding sub-problem, the primary path and protecting cycle of the request is found, respectively. First, the cycle finding method is described, then the path finding method.

5.1 Cycle Finding

After finding the primary path of the request, the protecting cycle should be determined to protect the path. For the cycle finding sub-problem, two methods are proposed in this article, they are *Fixed Cycle Finding* (FCF) and *Expandable Cycle Finding* (ECF). The set C of candidate cycles are pre-computed (by performing the *cycle-finding algorithm* [11]) for quick searching and without emulating all cycles. Cycles in C are sorted in increasing order according to the length of cycles.

5.1.1 Fixed cycle finding

In the FCF method, the FSs allocated to the cycles are fixed, that is, no FS expansion (or contraction) is allowed. This is the method used in previous study [7]. The currently allocated cycles can be used to protect new lightpath, only if the number of allocated FSs of the cycle can meet the requirement after adding the lightpath into the DRS of the cycle, which passes through the end-nodes of the lightpath. Moreover, if any lightpath protected by the cycle were released, no FS contraction is performed. Obviously, this may lead to resource waste since only partial FSs are used for protecting. The FSs allocated to the cycle can be completely released, if there is no lightpath protected by the cycle.

Let set $C_{s_i d_i} \subset C$ and $C^c_{s_i d_i} \subset C^c$ be the pre-computed set of candidate cycles currently deployed set of cycles, which pass through end-nodes s_i and d_i. Cycles in C are arranged in increasing order according to the length of cycles; if a tie, the cycle with higher protection efficiency (PE) is selected. The PE of a cycle $c_k \in C^c_{s_i,d_i}$ is defined as $PE(c_k) = \sum_{\forall p \text{ protected by } c_k} FS(p)/(((FS(c_k) + \varepsilon)|c_k|)$, where $|c_k|$ is the length of c_k, ε is a small positive constant to avoid zero-denominator, $FS(p)$ and $FS(c_k)$ is the number of FSs allocated on path p and cycle c_k, respectively. Cycles in C_{s_i,d_i} are arranged in increasing order according to the length of cycles.

In FCF, for a lightpath of a new request, the currently deployed cycles ($C^c_{s_i,d_i}$) are examined first. The first cycle found in $C^c_{s_i,d_i}$ is selected as the protecting cycle and the remaining cycle in $C^c_{s_i,d_i}$ are not examined, since the cost of the cycle in $C^c_{s_i,d_i}$ is zero. If the protecting cycle cannot be found in $C^c_{s_i,d_i}$, then cycyles in C_{s_i,d_i} are examined.The cycle in C_{s_i,d_i} with minimal length and can be allocated in EON is selected as the protecting cycle. The details of the FCF are described in Algorithm 1. For a primary lightpath, finding a protecting cycle by performing the FCF can be done in $O(D|V||C^c_{s_i,d_i}| + B|V||C_{s_i,d_i}|)$, where D is the maximal size of DRSs of all cycles in $C^c_{s_i,d_i}$.

Algorithm 1: Fixed Cycle Finding (FCF)

Input: $G(V, E)$, (s_i, d_i, f_i), primary lightpath p_a, $C^c_{s_i,d_i}$, C_{s_i,d_i};
Output: cycle c_k and fs_k;
while ($C^c_{s_i,d_i} \neq \emptyset$) **do**
    Select and remove a cycle c_k from $C^c_{s_i,d_i}$
    Check whether path p_a is link-disjoint or protection compatible to all current lightpaths protected by the cycle c_k and cycle c_k can be used to protect p_a without expansion.
    if (**true**) **then**
        return c_k and $fs_k = 0$;
    end if
end while
while ($C_{s_i,d_i} \neq \emptyset$) **do**
    Select and remove a cycle c_k from C_{s_i,d_i}.
    Check if there are free FSs can be allocated for the cycle c_k.
    if (**true**) **then**
        return c_k and $fs_k = f_i$;
    end if
end while
return ∞;

5.1.2 Expandable cycle finding

In the ECF method, the allocated FSs of the current cycles can be expanded or contracted, that is, just enough FSs are allocated to the cycle to protect the lightpaths of the respective DRS. If the lightpath protected by the cycle is released, not only the FSs of the lightpath are released, but the required number of FSs for the cycle is re-checked and contracted if possible. Moreover, if new lightpath is added into the cycle, but the required number of FSs for the cycle is re-checked and expanded if need. The side (or direction) of expansion/contraction is higher index first.

The cycle can be used to protect the new lightpath, if the number of allocated FSs of the cycle can meet the requirement of the lightpath or can be expanded without affecting other cycles. Moreover, current cycles are allowed to move to another index of FSs and expand to protect the primary path, if possible. In ECF, the cost of the protecting cycle is the product of the length of the cycle and the required extra FSs (denoted as fs_k) of the cycle to protect new lightpath. If the cycle c_k in C_{s_i,d_i} is selected to protect the primary, to speed up the searching process, only the remaining cycles with the same length as cycle c_k a are examined. The details of the ECF are described in Algorithm 2. For a primary lightpath, finding the minimal cost expandable cycle can be done in $O(DB|V||C^c_{s_i,d_i}| + B|V||C_{s_i,d_i}|)$, were D is the maximal size of DRSs of cycles in $C^c_{s_i,d_i}$.

Algorithm 2: Expandable Cycle Finding (ECF)

Input: $G(V, E)$, (s_i, d_i, f_i), primary lightpath p_a, $C^c_{s_i,d_i}$, C_{s_i,d_i};
Output: cycle c_k and fs_k;
found = **false**;
while ($C^c_{s_i,d_i} \neq \emptyset$) **do**
  Select and remove a cycle c_k from $C^c_{s_i,d_i}$.
  Check whether path p_a is link-disjoint or protection compatible to all current lightpaths protected by the cycle c_k and cycle c_k can be used to protect p_a without expansion.
  if (**true**) **then**
    return c_k and $fs_k = 0$;
  else if ((c_k can be expanded to protect p_a) **and** ($fs_k < f_i$)) **then**
    Let c_{best} be the cycle in $C^c_{s_i,d_i}$ with minimal expansion, update c_{best} to c_k and $cc_{best} = |c_k| \times fs_k$ $(0 < fs_k < f_i)$ if need, *found* = **true**.
  end if
end while
cf = **false**;
while ($C_{s_i,d_i} \neq \emptyset$) **do**
  Select and remove a cycle *ck* from C_{s_i,d_i}.
  if ((*cf* == **true**) **and** ($|c_k| >$ *min len*)) **then**
    break;
  end if
  Check if there are free FSs can be allocated for the cycle c_k.
  if (**true**) **and** ($|c_k| \times fs_k < cc_{best}$) **then**
    update c_{best} and cc_{best} if need; *found* = **true**, *min_len* = $|c_k|$ and *cf* =**true**;

```
    end if
end while
if (found) then
    return c_best and fs_k;
else
    return ∞;
end if
```

5.2 Path Finding

To find the routing path of the given connection request, two path routing methods are used, they are *Alternate Routing* (AR) and *Dynamic Routing* (DR).

5.2.1 Alternate routing

In the *Alternate Routing method*, first, the set of K shortest paths for the connection request is pre-computed as the candidate paths. Then, candidate paths are selected and examined in increasing order according to the length of the path, and the first-fit FS assignment method is used to determine the assigned FSs of the lightpath. Let set P_{s_i,d_i} be the set of candidate paths, for the connection request (s_i, d_i, f_i), the routing path p_a is selected from the set P_{s_i,d_i}. If the lightpath p_a cannot be allocated, then the cost w_a of working path p_a is set to infinity ($w_a = \infty$). If the lightpath p_a can be established, then the cost w_a is set to $|p_a| \times f_i$, where $|p_a|$ is the length of the path p_a.

After finding the working path of the request, the protecting cycle for the path p_a is examined. It is worth noting that a cycle $c \in C$ can be deployed many copies and each of which can be allocated with different number FSs or different FS index. The pair $\{p_{best}, c_{best}\}$ is used to keep the current minimum cost pair of the primary path and the protecting cycle. We use a cost$_{best}$ to store the minimal (or best cost) by summing up the working cost and backup cost. And, define $PE_{ak} = w_a + b_k$, if path p_a is selected to route the request with cost $w_a = |p_a| \times f_i$ and cycle c_k is selected to protect the path p_a with cost $b_k = |c_k| \times fs_k$.

For the FCF method for the cycle c_k in $C^{c}_{s_i,d_i}$, it is removed and examined whether all lightpaths protected by cycle c_k are link-disjoint or protection compatible with the path p_a. If true, return the cycle c_k and the value of fs_k is set to 0, the cost of b_k is set to 0. For the ECF method, for the protection compatible cycle c_k in $C^{c}_{s_i,d_i}$, the minimal number of required FSs (denoted as fs_k) should be expanded to protect path p_a is computed. If cycle c_k cannot be expanded to protect path p_a, the value of fs_k is set to infinity. If $c_k \in C_{s_i,d_i}$ is a new cycle, then fs_k is set to f_i. The b_k is the backup cost of the cycle $fs_k \times |c_k|$, where fs_k ($0 \le fs_k \le f_i$) is the number of required extra FSs for protecting new path.

After examining all possible cases ($|K| \times |C' = C^{c}_{s_i,d_i} \cup C_{s_i,d_i}|$), the minimal cost pair (p_a, c_k) is selected. If a tie, the pair with minimal fragmentation $FR(p_a, c_k)$ is selected. The fragmentation ratio [1] of the network is defined as $\sum_{\forall e_l \in E} MaxBlock(e_l) / \sum_{\forall e_l \in E}(B - \sum_{j=1}^{B} b_l(j))$, where $MaxBlock(e_l)$ is the maximum size of available contiguous slots in e_l. The details of the Alternate Routing Algorithm are described in Algorithm 3. In AR, the all-pairs candidate paths set can be constructed and preprocessed in $O(|E||V|^2 + |V|^3\log|V| +$

$k|V|^2$). For a given primary path, find the allocated FSs for the path can be done in $O(|V|B)$. And, finding the fixed protecting cycle can be done $O(D|V||C^c_{s_i,d_i}| + B|V||C_{s_i d_i}|)$. Thus, the time complexity of the AR_FC algorithm is $O(KD|V||C^c_{s_i,d_i}| + KB|V||C_{s_i d_i}|)$. Finding the minimal cost expandable cycle can be done in $O(BD|V||C^c_{s_i,d_i}| + B|V||C_{s_i d_i}|)$. The time complexity of the AR_EC algorithm is $O(KBD|V||C^c_{s_i,d_i}| + KB|V||C^c_{s_i,d_i}|)$.

Algorithm 3: Alternate Routing (AR)

Input: $G(V, E)$, (s_i, d_i, f_i), $P = \{P_{s_i,d_i} | \forall s_i, d_i \in V\}$, C^c, C_{s_i,d_i};
Output: lightpath p_i for (s_i, d_i, f_i), C^c;
$cost_{best} = \infty$, $p_{best} = null$, and $c_{best} = null$.
for all $(p_a \in P_{s_i,d_i})$ **do**
  for the selected path p_a, check whether it can be allocated on the network;
  if (**false**) **then**
    $w_a = \infty$;
  else
    temporarily allocate FSs for the path p_a;
    compute $w_a = |p_a| \times f_i$ and construct set $C' = C^c_{s_i,d_i} \cup C_{s_i,d_i}$);
    for all $(c_k \in C')$ **do**
      Check whether all current lightpaths protected by the cycle c_k are protection compatible with the path p_a.
      if (**true**) **then**
        Compute the minimal number of required extra FSs (fs_k) for the cycle c_k should be allocated (or expanded if possible) to protect path p_a.
      else
        $fs_k = \infty$;
      end if
      compute $b_k = fs_k \times |c_k|$, $PE_{ak} = w_a + b_k$;
      if $(PE_{ak} < cost_{best})$ **then**
        update $cost_{best}$ as PE_{ak}, $p_{best} = p_a$, and $c_{best} = c_k$;
      else if $(PE_{ak} == cost_{best})$ **then**
        if $(FR(p_a, c_k) < FR(p_{best}, c_{best}))$ **then**
          update $cost_{best}$ as PE_{ak}, $p_{best} = p_a$, and $c_{best} = c_k$;
        end if
      end if
    end for
  end if
  release the resources allocated to the path p_a;
end for
if $(cost_{best} < \infty)$ **then**
  Allocate FSs for the path p_{best} and add p_{best} to the DRS set of cycle c_{best}.
  Allocate or expand the required FSs for the cycle c_{best}. If c_{best} is a new cycle, add c_{best} to C^c, and **return success**.
else
  block the request and **return**;
end if

5.2.2 Dynamic routing

In this subsection, the details of the Dynamic Routing (DR) are described. In the Dynamic Routing method, the routing path and assigned FSs are determined dynamically according to the current network status. For an arrived connection request (say (s_i, d_i, f_i)), a path from s_i to d_i on current network is on-line computed to route the request. The primary path can be obtained by performing the LG-RSA algorithm [2]. The candidate set of paths is generated by performing the K-shortest path algorithm on the layered graphs $G^j(V^j, E^j)$, $j = 1, 2, ..., B - f_i + 1$. The jth layered graph is denoted as $G^j(V^j, E^j)$, where $V^j = V$ and $E^j = \{e_l \,|\, B_{lj} = \sum_{z=j}^{j+f_i+1} b_l(z) = 0 \text{ and } e_l \in E\}$. On $G^j(V^j, E^j)$, the link $e_l \in E_j$ represents that there are free continuous FSs within jth to $(j + f_i - 1)$th on the link $e_l \in E$ for the connection request. If a lightpath can be found on $G^j(V^j, E^j)$, it means that the lightpath can be allocated on network G and the starting index of FSs is j and for f_i continuous FSs.

If a routing path can be found, each candidate cycle in $C'(= C^c_{s_i,d_i} \cup C_{s_i,d_i})$ was examined whether they can be used to protect the lightpath. If the protecting cycle can be found, then the lightpath is added to the DRS of the cycle and return. The details of the Dynamic Routing algorithm are described in Algorithm 4. In ECF, if the cycle without FSs expansion cannot be found, then the existing expandable cycle, which can be used to protect the lightpath and with minimal expansion is selected and used to protect the lightpath. To expand allocated FSs of the protecting cycle, the free FSs on the lower side is expanded first, then the upper side. Moreover, there is no current allocated cycles or lightpaths are affected by this expansion. If there is no existing cycle can be used to protect the lightpath, then a new cycle is found to protect the lightpath if possible.

For a given request $r = (s_i, d_i, f_i)$, the layered graph can be constructed in $O(f_i \times |E|)$, and the K-shortest paths on the layered graph can be computed in $O(|E| + |V| \log |V| + K)$. All possible paths can be found is $O(B \times (f_i \times |E| + |E| + |V| \log|V| + K)) = O(Bf_i|E| + B|V|\log |V| + BK))$.

For each primary path, finding the fixed protecting cycle can be done in $O(D|V||C^c_{s_i,d_i}| + B|V||C_{s_i d_i}|)$. The time complexity of the DR_FC algorithm is $O(Bf_i|E| + BKD|V||C^c_{s_i,d_i}| + B^2K|V||C_{s_i d_i}|)$. For each primary path, finding the minimal cost expandable cycle can be done in $O(DB|V||C^c_{s_i,d_i}| + B|V||C_{s_i d_i}|)$. The time complexity of the DR_FC algorithm is $O(Bf_i |E| + KB \times (DB|V||C^c_{s_i,d_i}| + B|V||C_{s_i d_i}|)) = O(Bf_i|E| + KDB^2|V||C^c_{s_i,d_i}| + KB^2|V||C^c_{s_i,d_i}|)$.

Algorithm 4: Dynamic Routing (DR)

```
Input: G(V, E), (s_i, d_i, f_i), C^c, C
Output: lightpath p, C^c
cost_best = ∞, p_best = null, and c_best = null.
for all (j = 1, 2, ..., B − f_i + 1) do
    construct the layered graph G^j(V^j, E^j);
    Perform K-shortest paths algorithm on G^j(V^j, E^j) to find the set of candidate paths
    is P_j = {p_j1, p_j2, ..., p_jK}.
    for all (z = 1, 2, ..., K) do
        compute w_jz = |p_jz| × f_i and construct C′ = C^c_{s_i,d_i} ∪ C_{s_i,d_i};
        for all (c_k ∈ C′) do
```

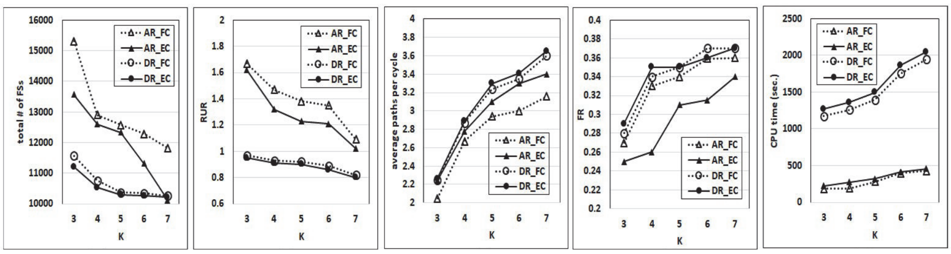
Check whether all current lightpaths protected by the cycle c_k are protection compatible with the path p_{jz}.
if (**true**) **then**
Compute (fs_k) the minimal number of required FSs for the cycle c_k should be expanded to protect path p_{jz}.
else
$fs_{jz} = \infty$;
end if
Compute $bk = fs_i \times |c_k|$, $PE_{jzk} = w_{jz} + b_k$.
if ($PE_{jzk} < cost_{best}$) **then**
Update $cost_{best}$ as PE_{jzk}, $p_{best} = p_{jz}$, and $c_{best} = c_k$.
else if ($PE_{jzk} == cost_{best}$) **then**
if ($FR(p_{jz}, c_k) < FR(p_{best}, c_{best})$) **then**
Update $cost_{best}$ as PE_{jzk}, $p_{best} = p_{jz}$, and $c_{best} = c_k$.
end if
end if
end for
end for
end for
if ($cost_{best} < \infty$) **then**
Allocate FSs for the path p_{best} and add p_{best} to the DRS set of cycle c_{best}.
Allocate or expand the required FSs for the cycle c_{best}, add c_{best} to C^c if c_{best} is a new cycle and **return success**.
else
Block the request and **return**.
end if
Block the request and **return**.

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed heuristic algorithms, these algorithms were implemented and applied to solve test examples, and the results are reported in this section. The implementation was conducted in C++. All the simulations were run on a notebook computer with an Intel Core i7-4270HQ CPU 2.60 GHz and 16GB RAM and with Windows 10 operating system. The physical network used for simulations is shown in Fig. 1 (b), which with 17 nodes, 63 edges and 400 FSs for each edge. Several performance criteria are evaluated: (1) total number of FSs allocated to primary and backup lightpaths; (2) CPU time; (3) Fragmentation Ratio (FR); (4) the average number of lightpaths protected by a cycle; and (5) resource utilization ratio; (RUR): the total number of FSs used by the protecting cycles to that of the primary lightpaths.

For static case, connection requests are generated with equal probability for all possible node pairs. The number of required FSs for the request is within 1-10. The default number of candidate paths for each request in Alternate Routing and Dynamic Routing is $K = 3$. Several sets which are with different numbers of candidate cycles per node-pair are generated, at least 2-6 candidate cycles for each node-pair are selected from all possible cycles in increasing order.

Four possible algorithms are AR_FC, AR_EC, DR_FC and DR_EC. The notation 'EC' stands for 'expandable cycle allocation' and 'FC' for 'fixed cycle allocation', respectively. The simulation results for 600 connection requests for different values of *K* of the proposed algorithms are shown in Fig. 3, where *K* is the maximal number of candidate paths pre-computed for each request in AR and each layered graph in DR. In this simulation, for each node-pair, at least two protecting cycles are generated as the candidate cycles in set *C*. In Fig. 3 (a), as the value of *K* increases, the total number of FSs required for these methods decrease. The EC method can get a smaller amount of FSs than that of the FC method. The DR method can get better results than that of the AR method. The AR method is more responsive to the value of *K* than the DR method, AR can find better results as *K* increases. Fig. 3 (b) shows that RUR decreases as the value of K increases, for the DR method, the RUR value is lower than 1. Figs. 3 (c) and (d) shows that average lightpaths protected by a cycle and FR increases as the value of K increases, respectively. DR method can get better FR than that of the AR method. In Fig. 3 (e), as the value of *K* increases, the CPU time of the algorithms increases. The DR method is more time-consuming than AR method since the number of candidate paths of the DR method is *B*×*K* far more than that of the AR method.

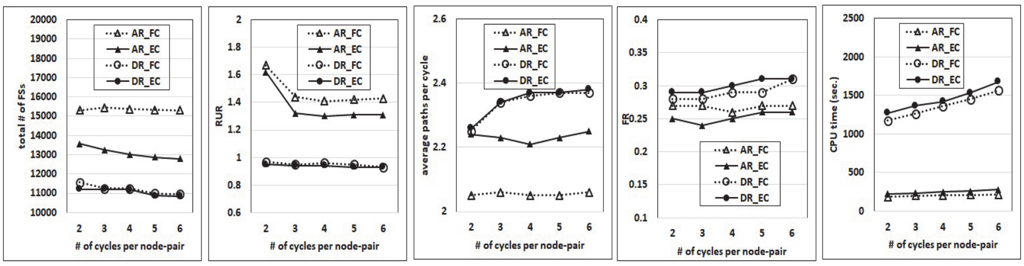


(a) (b) (c) (d) (e)

Fig. 3. Simulation results for different values of *K* (a) total number of FSs; (b) RUR; (c) the average number of lightpaths protected by a cycle; (d) FR; (e) CPU time in seconds.

Several sets which are with different numbers of candidate cycles per node-pair are generated, at least 2-6 candidate cycles for each node-pair are selected from all possible cycles in increasing order. The simulation results for 600 connection requests for different sets of candidate cycles of the proposed algorithms are shown in Fig. 4. Fig. 4 (a) shows results of the total number of FSs allocated for different sets of cycles. As the number of candidate cycles increases, the total number of FSs decreases a little. In Fig. 4 (a), the DR EC can get the minimal number of total FSs and DR FC is the second best method. Fig. 4 (b) shows that the RUR value decrease a lot as the set changes from with at least 2 to 3 candidate cycles for AR method, but there is almost no change in other cases. This result also shows that DR methods can get a better result than AR methods.

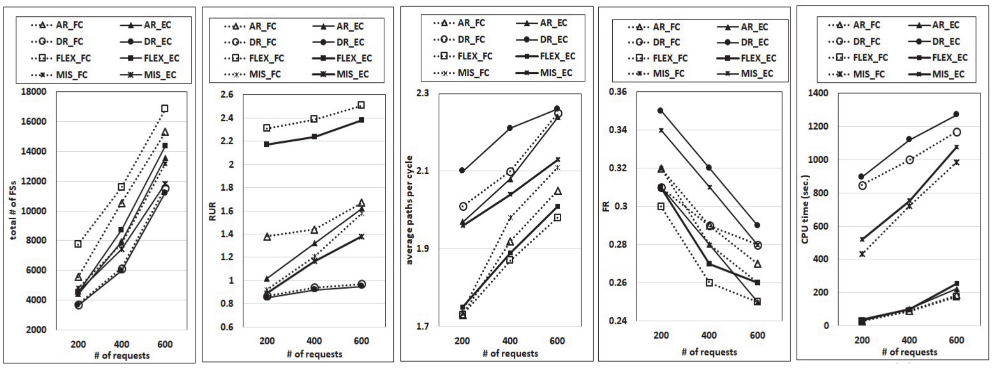
Fig. 4 (c) shows that the average number of lightpaths protected by cycle increases for DR method, as the number of candidate cycles increases. Fig. 4 (d) shows that the FR increases a little, as the number of candidate cycles increases. Fig. 4 (e) shows that the CPU time increases a lot for the DR method, as the number of candidate cycles increases.



(a) (b) (c) (d) (e)

Fig. 4. Simulation results for 600 connection requests (a) total number of allocated FSs; (b) RUR; (c) the average number of lightpaths protected by a cycle, (d) FR, (e) CPU time in seconds.

To know the efficiency of the proposed methods, four extra algorithms in the literature are also implemented for comparison. The method FLEX stands for the algorithm proposed in [6] and the MIS stands for the best method MIS-FIPP algorithm proposed in [7]. The FLEX and MIS methods were also extended by considering expandable cycles allocation and denoted as FLEX EC and MIS EC, respectively. The original version of the FLEX and MIS methods are denoted as FLEX FC and MIS FC, respectively. Fig. 5 shows the comparison of these methods with different numbers (200, 400, 600) of connection requests. Fig. 5 (a) shows that the total number of FSs increase as the number of connection requests increases. The FLEX FC is the worst method and the DR EC is the best method. After integrating the EC method into FLEX and MIS, the total number of allocated FSs can be reduced. Fig. 5 (b) shows that the FLEX FC gets the worst RUR result and DR EC gets the best RUR result. Fig. 5 (c) shows that the average number of paths protecting by cycle increases as the number of connection requests increases for the AR method. Fig. 5 (d) shows that the FR decreases as the number of connection requests increases for most of the cases. Fig. 5 (e) shows that the CPU time increases as the number of connection requests increases. The FLEX is the fastest method and the DR is the most time-consuming method.



(a) (b) (c) (d) (e)

Fig. 5. Simulation results for different numbers of connection requests (a) total number of allocated FSs; (b) RUR; (c) the average number of lightpaths protected by a cycle; (d) FR; (e) CPU time in seconds.

7. CONCLUSIONS

For the given network and the set of connection requests, the lightpaths used to route the connection requests and the protecting cycles, which are used to protect the connections against single-link failure are found. The FIPP *p*-cycles protection scheme is used for lightpath protection. In this paper, a dynamic scheme for FS allocation in EONs is considered to increase the efficiency of the backup resources for supporting the FIPP *p*-cycles protection. Two heuristic algorithms AR and DR have been proposed to route the connection requests on EONs. The proposed algorithms have been evaluated by extensive simulations and the results show that through introducing the expandable cycle allocation and releasing link-disjoint constraint, the spectrum efficiency can be improved.

REFERENCES

1. G. Zhang, M. D. Leenheer, A. Morea, and B. Mukherjee, "A survey on OFDM-based elastic core optical networking," *IEEE Communications Survey and Tutorials*, Vol. 15, 2013, pp. 65-87.
2. X. Liu, L. Gong, and Z. Zhu, "Design integrated RSA for multicast in elastic optical networks with a layered approach," in *Proceedings of IEEE Global Telecommunications Conference*, 2013, pp. 2346-2351.
3. G. Shen, H. Guo, and S. K. Bose, "Survivable elastic optical networks: survey and perspective," *Photonic Network Communications*, Vol. 31, 2016, pp. 71-87.
4. A. Kodian and W. D. Grover, "Failure-independent path-protecting *p*-cycles: efficient and simple fully pre-connected optical path protection," *Journal of Lightwave Technology*, Vol. 23, 2005, pp. 3241-3259.
5. R. Asthana, Y. Singh, and W. Grover, "*P*-cycles: an overview," *IEEE Communications Surveys Tutorials*, Vol. 12, 2010, pp. 97-111.
6. H. M. N. S. Oliveira and N. L. S. da Fonseca, "Algorithm for FIPP *p*-cycle path protection in flex grid networks," in *Proceedings of IEEE Global Communications Conference*, 2014, pp. 1278-1283.
7. X. Chen, S. Zhu, L. Jiang, and Z. Zhu, "On spectrum efficient failure-independent path protection *p*-cycle design in elastic optical networks," *Journal of Lightwave Technology*, Vol. 33, 2015, pp. 3719-3729.
8. H. M. N. S. Oliveira and N. L. S. da Fonseca, "Algorithm for protection of space division multiplexing elastic optical networks," in *Proceedings of IEEE Global Communications Conference*, 2016, pp. 1-6.
9. H. M. N. S. Oliveira and N. L. S. da Fonseca, "The minimum interference *p*-cycle algorithm for protection of space division multiplexing elastic optical networks," Technical Report, No. IC-17-04, Relatorio Tecnico, 2017.
10. S. Zhu, S. Meng, Q. Bao, X. Chen, and Z. Zhu, "Availability-aware service provisioning in EONs: How efficient will FIPP-*p*-cycles be?" in *Proceedings of Optical Fiber Communications Conference*, 2016, pp. W2A.51.
11. C. Liu and L. Ruan, "Finding good candidate cycles for efficient *p*-cycle network design," in *Proceedings of the 13th International Conference on Computer Communications and Networks*, 2004, pp. 321-326.

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.