

2006

## A Hybrid p-Cycle Search Algorithm for Protection in WDM Mesh Networks

Kungmang Lo  
*Edith Cowan University*

Daryoush Habibi  
*Edith Cowan University*

Alexander Rassau  
*Edith Cowan University*

Quoc Viet Phung  
*Edith Cowan University*

Hoang Nguyen  
*Edith Cowan University*

*See next page for additional authors*

Follow this and additional works at: <https://ro.ecu.edu.au/ecuworks>

 Part of the [Engineering Commons](#)

---

[10.1109/ICON.2006.302637](https://ro.ecu.edu.au/ecuworks/1947)

This is an Author's Accepted Manuscript of: Lo, K. , Habibi, D. , Rassau, A. M., Phung, Q. , Nguyen, H. N., & Kang, B. K. (2006). A Hybrid p-Cycle Search Algorithm for Protection in WDM Mesh Networks. Proceedings of IEEE International Conference on Networks. (pp. 262-267). Singapore. IEEE. Available [here](#)

© 2006 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This Conference Proceeding is posted at Research Online.  
<https://ro.ecu.edu.au/ecuworks/1947>

---

**Authors**

Kungmang Lo, Daryoush Habibi, Alexander Rassau, Quoc Viet Phung, Hoang Nguyen, and Byung Kang

# A Hybrid $p$ -Cycle Search Algorithm for Protection in WDM Mesh Networks

Kungmeng Lo, Daryoush Habibi, Alexander Rasan, Quoc Viet Phung, Hoang Nghia Nguyen, and ByungKyu Kang  
 COMMUNICATIONS RESEARCH GROUP, SCHOOL OF ENGINEERING AND MATHEMATICS  
 EDITH COWAN UNIVERSITY  
 PERTH, AUSTRALIA

**Abstract**— $p$ -Cycle is a type of shared link protection for survivable wavelength-division multiplexing (WDM) mesh networks.  $p$ -Cycle not only retains ring-like restoration speeds, but also achieves capacity efficiency in mesh networks. However, finding the optimal set of  $p$ -cycles to protect all traffic demands within a reasonable response time is difficult. This is particularly true with dense meshes or large networks, because the number of candidates is huge.

Generally,  $p$ -cycles are determined by using either Integer Linear Programming (ILP) or specifically designed heuristic algorithms. However, both methods need a set of efficient candidate cycles to tradeoff between the computational time and the optimality of solutions. For this reason, constructing an efficient set of candidate  $p$ -cycles is crucial and imperative. In this paper, we propose the Span-weighted Cycle Searching (SCS) algorithm to generate and select an adequate number of  $p$ -cycles to minimize the spare capacity, while achieving 100% restorability, within low computational complexity.

**keywords:**  $p$ -cycle, wavelength-division multiplexing (WDM), survivability,

## I. INTRODUCTION

*Survivability* in optical networks is an important issue, particularly in Wavelength-Division multiplexing (WDM) networks where failure of a fiber link or node may cause huge data losses. In addition, if the protection can be done in the optical domain, the upper level protocols, i.e. IP or ATM, do not need to be aware of changes in the underlying network topology. The aim of survivability is to restore failed traffic flows via alternative backup routes. The rerouting process may be performed between end nodes of affected traffic connections, referred to as *Path Protection*, or between two end nodes of the failed link, referred to as *Link Protection*. Research has shown that path protection offers better capacity utilization but link protection offers faster restoration [1].

The concept of  $p$ -cycles for protection was proposed by Grover and Stamatelakis in 1998 [2] [3] [4].  $p$ -Cycles is a type of pre-configured protection in mesh networks.  $p$ -Cycles protection has the fast recovery speed of ring restoration and the capacity efficiency of mesh protection. Moreover,  $p$ -cycles support independent shortest-path routing for traffic demands, without constraints arising from the placement of the protection structures. This method addresses the speed issues of mesh-based restoration, because restoration is handled individually by the two nodes of the failed span.

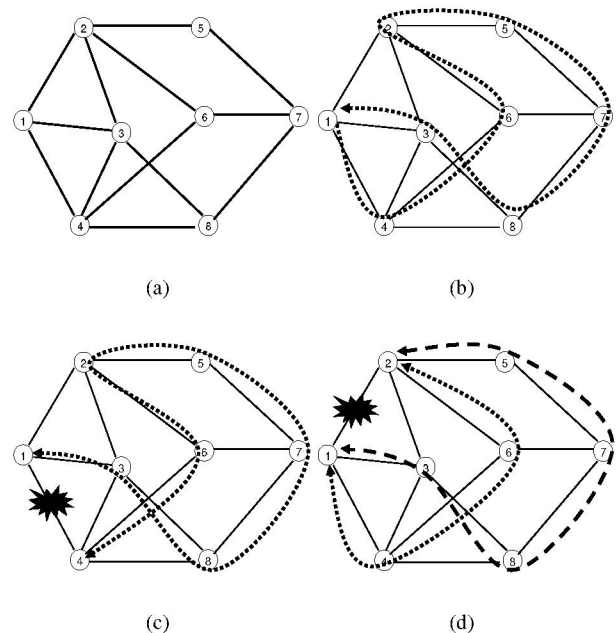


Fig. 1. An 8n13s network (a) with one  $p$ -cycle (b), which can restore one on-cycle span (c) and two straddling spans (d)

$p$ -Cycles can be configured efficiently over a mesh-structure network [5]. An individual  $p$ -cycle can restore one unit of working capacity on every on-cycle span, and two units of working capacity on every straddling span. A straddling span for a cycle is a span whose two end-nodes are on the cycle, but the span itself is not part of that cycle. In Figure 1(b) we note that the  $p$ -cycle 1-4-6-2-5-7-8-3-1 provides a single restoration path for 8 span failures as shown in Figure 1(c), and two restoration paths for each of the five straddling spans 1-2, 2-3, 3-4, 4-8 and 6-7, as presented in Figure 1(d).

Some factors that influence the applicability of  $p$ -cycles include network size (or topology, configuration), total traffic loads and working capacity patterns, etc. A predictor tool to examine the potential efficiency of a  $p$ -cycle for protecting the working capacity is necessary. One such tool is a *a priori efficiency* [6], defined as:

$$AE(p) \triangleq \frac{\sum_{\forall l \in L} X_{p,l}}{\sum_{(\forall l \in L | X_{p,l}=1)} cost_l}$$

where  $l$  is a span in  $L$ ,  $X_{p,l}$  is the protection potential related to span  $l$  ( $X_{p,l} = 1$  when span  $l$  is an on-cycle span and  $X_{p,l} = 2$  if span  $l$  is a straddling span), and  $cost_l$  is the cost of capacity on span  $l$ . If the cost of each span  $l$  is set to a unit of hop, then the total cost of a cycle is the number of hops in it, and the case is referred to as the *non-cost-weighted* or *logical a priori efficiency* [7], which can be expressed as:

$$AE(p) = \frac{2 \times |S_{st,p}| + |S_{on,p}|}{|S_{on,p}|} \quad (1)$$

where  $S_{st,p}$  is the number of straddling spans, and  $S_{on,p}$  is the number of on-cycle spans. In theory, a cycle with a large AE value will have a high potential for efficient protection of working capacity. However, the efficiency of a cycle also needs to take into account the working capacity distribution.

Existing approaches for solving the  $p$ -cycles problem include some ILP and heuristic algorithms, both of which require a set of candidate  $p$ -cycles to be pre-computed. Finding the optimal solution for small or medium size networks is quick and easy because of the small number of available cycles. For example, the network topology in Figure 1 which has 8 nodes, 13 spans, and an average nodal degree of  $\bar{d} = 3.25$ , has a total of 40 possible cycles, and the NSFNet topology (14 nodes, 21 spans,  $\bar{d} = 3$ ) has a total of 139 cycles [8]. However, the number of candidate  $p$ -cycles may be huge in dense or large networks, posing a difficult problem to solve for both approaches [7]. For example, there are 3531 possible cycles in the COST239 topology (11 nodes, 26 spans,  $\bar{d} = 4.73$ ) [6] [5], 8857 possible cycles in the Pan-European Optical Network (EON) topology (19 nodes, 38 spans,  $\bar{d} = 4$ ), and 7321 possible cycles in the USA topology (28 nodes, 45 spans,  $\bar{d} = 3.2143$ ) [7]. Thus finding a subset of candidate  $p$ -cycles to improve the performance is an important issue, and the objective is to minimize the total amount (or cost) of the spare capacity whilst retaining 100% restorability.

The rest of this paper is organized as follows. In section II we review the existing algorithms for finding candidate  $p$ -cycles and their performance. We then propose a Span-weighted Cycle Selection (SCS) algorithm, which is designed to compute an efficient and sufficient subset of  $p$ -cycles in Section III. In section IV we evaluate the performance of the SCS algorithm using simulation over some arbitrary mesh topologies. Finally, we conclude this paper in section V.

## II. RELATED WORK REVIEW

Research into  $p$ -cycles for network protection has generated recent interest amongst the research community. Zhang *et al.* propose a Cycle-based Rerouting Scheme (CRS) for link protection [9] [10]. In the CRS, each link has a small set of candidate rerouting paths. Then, a Straddling Link Algorithm (SLA) is proposed to generate a small subset of  $p$ -cycles of a network graph. A cycle could be combined with a pair of node-disjoint paths between the two end-nodes of the link. This link is then a straddling link for the cycle. The SLA algorithm is simple and fast for constructing candidate cycles,

but those candidates are insufficient to satisfactorily minimize the spare capacity.

Another scheme to generate candidate  $p$ -cycles, which iteratively select a set of cycles using a heuristic approach to minimize the spare capacity is introduced by Doucette *et al.* [7]. The candidates are built in two stages. First, a set of primary cycles is generated using the SLA [9] [10]. Then the large cycles are derived by the proposed ‘Add’, ‘Join’, or ‘Grow’ algorithms. It is reported that the Grow algorithm can generate the most efficient candidate cycles. The heuristic approach for selecting cycles is known as Capacitated Iterative Design Algorithm (CIDA). Instead of an ILP model, cycles are iteratively chosen from the candidates and placed in the network to reduce the unprotected working capacity until all working capacities are protected. In CIDA, the right cycle is selected by calculation of the highest work-weighted efficiency  $E_w$ . The  $E_w(p)$  of a cycle  $p$  is computed as:

$$E_w(p) = \frac{\sum_{\forall l \in L} w_l \cdot S_{l,p}}{\sum_{(\forall l \in L | S_{l,p}=1)} cost_l} \quad (2)$$

where  $w_l$  is the amount of unprotected working capacity on span  $l$ .  $S_{l,p}$  is the number of protection channels related to  $l$  from the cycle  $p$ .  $S_{l,p} = 1$  if  $l$  is an on-cycle span, and  $S_{l,p} = 2$  if  $l$  is a straddling span.  $L$  is the set of spans in the network, and  $cost_l$  is the cost of a unit capacity on  $l$ . The idea of  $E_w$  is that a right cycle depends not only on the number of straddling spans and on-cycle spans, but also on the working capacities of those spans. The best performance is achieved by the CIDA-*Grow* algorithm, the redundancy is 10% different from the optimal value.

Liu and Ruan [11] propose the Weighted DFS-based Cycle Search (WDCS) algorithm to find good candidate cycles. The candidate cycles are constructed by the Depth First Search (DFS) algorithm. The candidates consist of both high efficiency cycles and small cycles, so that both densely distributed and sparsely distributed working capacities can be efficiently protected by the candidate cycles. Basically, each span can be protected by at least one high efficiency cycle and two short cycles (one is to be the on-cycle span, the other is to be the straddling span). The number of protection cycles on each span can be increased by changing the input parameter  $k$ .

Another algorithm called Dynamic  $p$ -cycles Selection (DPS) is presented in [8]. Cycles are classified as sets of large cycles and small cycles, like the WDCS algorithm [11]. The candidate cycles are dynamically selected based on the network state (topology and working capacity pattern). Then the ILP model is used to select cycles which minimize the spare capacity. However, DSP selects large number of candidates leading to a high computation complex, although the performance is acceptable.

## III. NETWORK PROTECTION DESIGN

The objective of  $p$ -cycles in a static traffic environment is to find the minimum spare capacity, as well as allowances complete restoration of the working capacity. A set of *efficient* candidate cycles should be built in advance. The purpose of

efficient candidates is to balance the number of cycles and the performance. To do so, the Span-weighted Cycle Searching (SCS) algorithm is proposed. We firstly consider a set of cycles which contain large cycles ( $AE(p) > 1$ ) and elementary cycles ( $AE(p) = 1$ ). The large cycles are only searched from the span with the highest value (weight) of total nodal degree on its two end-nodes. The advantage of this approach is that the efficient cycles are only searched from the busiest span(s), rather than from every span. Whilst only the elementary cycles are found at every span. This can simplify the search process. Finally, we study a set of parameters to determine an adequate number of candidate cycles.

Three procedures are presented to fulfill the objective in this section. First, we provision the working traffic. The purpose is to minimize the total working capacity and also provide enough capacity for protection cycles. Next, a set of candidate  $p$ -cycles is generated using the SCS algorithm. Finally, the qualified cycles are computed using an ILP model. Our model is based on the following assumptions:

- The network topology is bi-connected and each fiber link is bidirectional.
- Each node has the same array of transmitters and receivers.
- Call requests are end-to-end connections.
- Only single link failures are considered.
- Full wavelength conversion is available at all nodes of the network.
- The traffic demands are known in advance.

To illustrate the SCS algorithm, we model the physical topology as a connected graph  $G(N, L)$ , where  $N$  is the set of wavelength routing nodes and  $L$  is the set of single-fiber links. The notations of the study model are:

#### A. Minimizing the Working Capacity

Given a set of traffic demands, we first provision the working capacity allocation to prevent undue traffic congestion occurring at any span, such that each span has enough spare capacity for  $p$ -cycles. The reason for this step is because the ILP model normally optimizes the working capacity on each span based on the shortest-path routing algorithm. This may lead to high working capacity on some spans, making it harder to find enough spare capacity on these spans for  $p$ -cycle protection. Therefore, the routing of the traffic demands has to be adjusted if a protecting set of  $p$ -cycles cannot be found. This situation can particularly arise in those spans that connect nodes with  $deg(n) = 2$ , e.g.,  $n_5$  with  $deg(n_5) = 2$  as shown in Figure 1(a). If the working capacity on spans  $l_{2,5}$  ( $w_{25}$ ) or  $l_{5,7}$  ( $w_{57}$ ) is more than half of the span capacity,  $c_l/2$ , then it is not possible to achieve 100% working capacity protection. The ILP model to obtain optimal solutions for the design of the working capacity is given below.

Minimize:

$$\sum_{l=1}^{|L|} \omega_l \quad (3)$$

$N = \{n_i\}$	the set of network nodes (e.g. OXCs or OADMs). $i \in \{1 \dots  N \}$ .
$L = \{l_j\}$	the set of optical fiber links in the network, where $j \in \{1 \dots  L \}$ .
$l_{u,v}^{deg=max}$	the two-end nodes $n_u$ and $n_v$ on span $l$ . span $l$ connects the node with the highest nodal degree.
$l^{deg=2}$	span $l$ connects the node with nodal degree 2.
$\omega_{i,j}$	the working capacity between node $n_i$ and $n_j$ .
$\omega_l$	the working capacity on the span $l$ .
$\omega_l^{deg=max}$	the working capacity on the span $l$ , which connects the highest nodal degree in the network.
$\omega_l^{deg=2}$	the working capacity on the span $l$ , which connects a node of degree 2.
$deg(n_i)$	the nodal degree at node $n_i$ . In a random mesh topology, $2 \leq deg(n_i) \leq N - 1$ .
$C$	the total capacity in a fiber span.
$c_l$	the number of capacity on the span $l$ .
$K$	the number of candidate routes for each connections. of hop numbers, $i \in \{1 \dots k\}$ ; $s, d \in \{1 \dots N\}$ , $s \neq d$ .
$Path_i^i$	the $i^{th}$ shortest path between the two-end nodes in link $l$ .
$D$	the number of traffic demands.
$T$	the set of given traffic demands in which $t_d$ , $d = [1 \dots D]$ is the volume of demand $d$ .
$p_{d,k}$	the $k^{th}$ candidate route of connection $d$
$\beta_{d,k}$	an indicator variable for candidate route $p_{d,k}$ .
$On\_cycle^l$	a set of small cycles ( $AE = 1$ ), which contains span $l$ and without any straddling links in them.
$AE\_cycle^l$	a set of large cycles ( $AE \geq 1$ ), where span $l$ is the straddling link in them.
$P$	the set of candidate $p$ -cycles, which is selected by the SCS algorithm.
$p_i$	the $i^{th}$ $p$ -cycle in $P$ , $i \in \{1 \dots  P \}$ .
$s_l$	the spare capacity on the span $l$ .

subject to:

$$\sum_{k=1}^K \beta_{d,k} = t_d \quad (4)$$

$$\omega_l = \sum_{d=1}^{|D|} \sum_{k=1}^K b_{d,k}^l \times \beta_{d,k}, \quad (5)$$

where,

$$b_{d,k}^l = \begin{cases} 1 & \text{if } p_{k,d} \text{ uses the span } l, \forall l \in L. \\ 0 & \text{otherwise} \end{cases}$$

$$\omega_l^{deg=2} \leq \frac{c_l}{2} \quad (6)$$

$$\omega_l^{deg>2} \leq c_l \quad (7)$$

The objective function (3) minimizes the total working capacity for the traffic demands. (4) ensures that all traffic demands can be carried on candidate routes. Sufficient capacity for working routes is provided in constraint (5). The capacity constraints in (6) and (7) attempt to avoid the problem of undue traffic congestion on each fiber link.

#### B. Span-weighted Cycle Selection (SCS) Algorithm

In this section, we detail the construction of the candidates using the Span-weighted Cycle Searching (SCS) algorithm. The SCS algorithm searches  $p$ -cycles based on the weight of spans in the network. Two classes of cycles are generated,  $On\_cycle$  for elementary cycles and  $AE\_cycle$  for large

cycles. Two tasks in SCS are to construct  $On\_cycle$  and  $AE\_cycle$  according to the following rules:

Task 1: Construct the  $On\_cycle$

- An  $On\_cycle^l$  is a set of elementary cycles on span  $l$ . All cycles in  $On\_cycle^l$  contain  $l$  itself and do not contain any straddling link ( $S_{st,p} = 0, AE(p) = 1$ ).
- $On\_cycle^l$  is computed for all spans of the network.
- $On\_cycle^l$  is organized in the ascending order of the hop length.
- the number of  $on$  is selected  $on^{th}$  cycle on  $On\_cycle^l$ .  $on$  is computed for the following condition:

$$on = \begin{cases} 0.4 \times |On\_cycle^l| & \text{if } l^{deg}=2 \\ 0.2 \times |On\_cycle^l| & \text{otherwise} \end{cases} \quad (8)$$

The parameters 0.4 and 0.2 are acquired based on empirical analysis.

- Store the  $On\_cycle^l(on)$  into  $On\_cycle, \forall l \in L$ .

Task 2: Construct the  $AE\_cycle$

- $AE\_cycle^l$  is computed if and only if span  $l$  has the highest value (weight) of total nodal degree on it's two end-nodes, eg. two set of  $AE\_cycle$  are constructed at spans  $l_{2,3}$  and  $l_{3,4}$  in Figure 1(a).
- For  $AE\_cycle^l$ , each cycle must contain at least one straddling link ( $S_{st,p} \geq 1, AE(p) > 1$ ), that is the span  $l$  itself.
- $AE\_cycle^l$  is then arranged in descending order of the value of *a priori efficiency* given in (1).
- The  $ae$  score of the selected  $ae^{th}$  cycle on  $AE\_cycle^l$  must satisfy the condition:

$$ae = |AE\_cycle^l| \times \alpha \quad (9)$$

where  $0 < \alpha \leq 1$ .  $\alpha$  is a *controlling factor*, and we use it to control the number of candidate cycles in our model. The value of  $\alpha$  depends on the size of the network.

- Store the  $AE\_cycle^l(ae)$  into  $AE\_cycle, \forall l^{deg}=max$ .

Finally,  $On\_cycle$  and  $AE\_cycle$  are joined together, and then stored in  $P$ . The pseudocode of the SCS is presented in Algorithm 1.

### C. Spare Capacity Optimization

The final procedure is to compute the optimal solution for the spare capacity. The objective is to minimize the total protection capacity, as given in (10).

Minimize:

$$\sum_{l=1}^{|L|} s_l \quad (10)$$

subject to:

$$s_l = \sum_{\forall p_i \in P} \delta_{p_i,l} \times p_i \quad (11)$$

$$\omega_l \leq \sum_{\forall p_i \in P} \sigma_{p_i,l} \times p_i \quad (12)$$

$$s_l \leq c_l - \omega_l \quad (13)$$

---

### Algorithm 1 Span-weighted Cycle Searching (SCS)

---

**Require:**  $Path_l^i, \forall i \in \{1 \dots K\}, \forall l \in L$

**Ensure:**  $On\_cycle, AE\_cycle$  and  $P$

$\forall l: On\_cycle^l \leftarrow \emptyset; \forall l^{deg}=max: AE\_cycle^l \leftarrow \emptyset$

**for**  $l = 1$  to  $|L|$  **do**

**for**  $i = 2$  to  $K$  **do**

**if**  $(l, Path_l^i)$ : node-disjoint paths **then**

$On\_cycle^l \leftarrow (l \cup Path_l^i)$

**end if**

**end for**

  Arrange the order of  $On\_cycle^l$  in ascending hop number

  Select  $On\_cycle^l(on)$  of cycles by (8)

$On\_cycle \leftarrow On\_cycle^l(on)$

**if**  $l \triangleq l^{deg}=max$  **then**

**for**  $i = 2$  to  $K - 1$  **do**

**for**  $j = i + 1$  to  $K$  **do**

**if**  $(Path_l^i, Path_l^j)$ : node-disjoint paths **then**

$AE\_cycle^l \leftarrow (Path_l^i \cup Path_l^j)$

**end if**

**end for**

**end for**

    Arrange the order of  $AE\_cycle^l$  in descending AE value

    Select  $AE\_cycle^l(ae)$  of cycles by (9)

$AE\_cycle \leftarrow AE\_cycle^l(ae)$

**end if**

**end for**

$P \leftarrow (On\_cycle, AE\_cycle)$

---

where variables  $p_i$  and  $s_l$  are non-negative integers. The constants  $\delta_{p_i,l}$  and  $\sigma_{p_i,l}$  are defined as:

$$\delta_{p_i,l} = \begin{cases} 1 & \text{if cycle } p_i \text{ overlies span } l, \forall l \in |L| \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma_{p_i,l} = \begin{cases} 2 & \text{if } l \text{ is a straddling span of cycle } p_i \\ 1 & \text{if } l \text{ is an on-cycle span of cycle } p_i \\ 0 & \text{otherwise} \end{cases}$$

Constraint (11) limits the spare capacity in each span. Constraint (12) ensures that sufficient capacity is available on p-cycles to protect working channels, and constraint (13) assures that the sum of the working and backup capacities is within the upper limit of the number of available wavelength channels on each span.

## IV. NUMERICAL RESULTS AND ANALYSIS

In this section, we first analyze the impact of different values for  $\alpha$  on the cycle generation using the SCS algorithm. Three test networks are implemented and they are: the 8n13s, the 14n21s (NSFNet) and 11n23s, which are shown in Figs. 1, 2(a) and 2(b), respectively. We then evaluate our model in the EON and COST239 networks using a suitable  $\alpha$  value based on our earlier simulations. Our computing platform is an IBM ThinkCentre PC, with an Intel Pentium IV 3.0-GHz processor,

1 GB of RAM running Windows XP. We assume each span has sets of the same capacity ( $C = 24$ ).

test networks are:  $l_{2,3}$  and  $l_{3,4}$  on 8n13s;  $l_{3,4}$ ,  $l_{4,7}$ ,  $l_{4,14}$ ,  $l_{9,10}$ ,  $l_{10,12}$ , and  $l_{10,13}$  on 14n21s; and  $l_{3,10}$  on 11n23s. Table II shows the numbers of candidates, this is generated by SCS for each  $\alpha$  value, and what percentage of the total cycles.

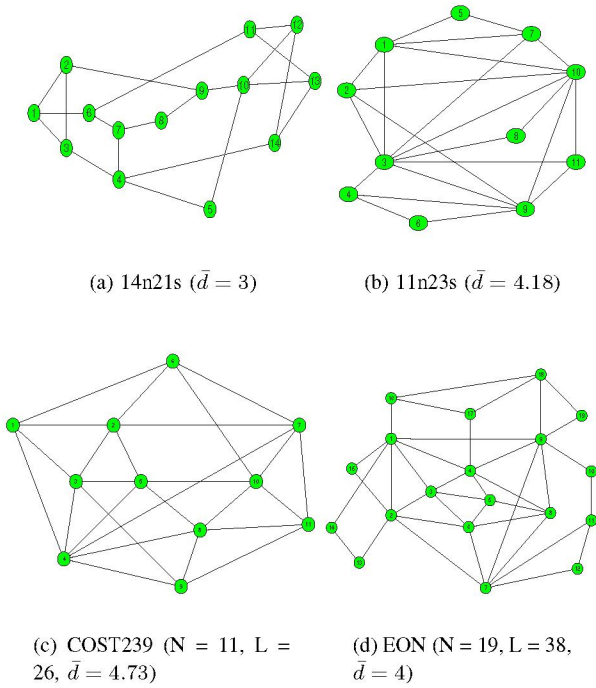


Fig. 2. The test network topologies

For a given traffic demand, we first provision the working traffic by (3) to (7). Then a set of candidate cycles is constructed by the SCS algorithm. Finally, the selected cycles are determined by (10) to (13). In order to compare our results with the optimal solutions, all cycles for the three test networks should be found. The total cycles can be built by using the modified SCS algorithm. That is, for the construction *On\_cycle*, we remove the constraints (8) and (9); and in the *AE\_cycle*, cycles are searched at every span. The total number of  $p$ -cycles and other parameters for the three test networks are shown in Table I.

TABLE I  
TOTAL CYCLES CONSTRUCTED BY THE SCS ALGORITHM

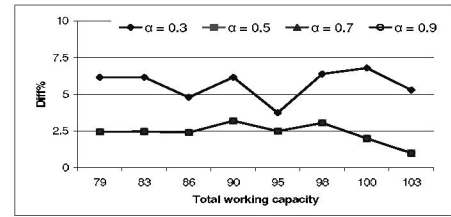
The test networks	8n13s	14n21s	11n23s
Total cycles	40	139	307
Average AE value	1.55	1.42	2.48
Average hop number	5.68	9.59	6.12
The value of $K$	8	22	41
<i>On_cycle</i>			
No. of oncycles	10	32	17
Average hop number	4.1	6.56	3
<i>AE_cycle</i>			
No. of AECycles	30	107	290
Average hop number	6.2	10.50	6.31
Average AE value	1.73	1.54	2.57

#### A. Analyzing the parameter $\alpha$

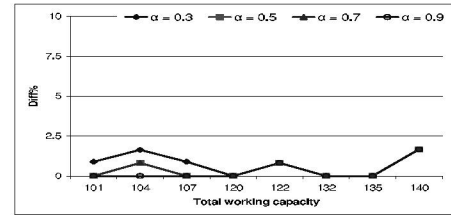
In this paper, the values of  $\alpha$  in (9) are taken to be 0.3, 0.5, 0.7 and 0.9. The spans for constructing *AEcycle* for the

TABLE II  
NUMBERS OF CANDIDATES AND THE RATE FOR THE TEST NETWORKS

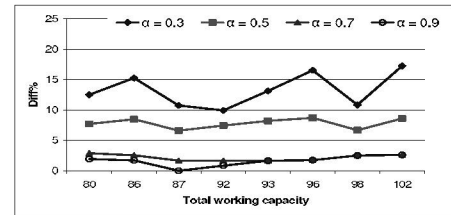
	$\alpha$	8n13s		14n21s		11n23s	
		No.	%	No.	%	No.	%
	0.3	12	30	42	30.2	78	25.4
	0.5	15	37.5	57	41	120	39.1
	0.7	19	47.5	71	51.1	162	52.8
	0.9	23	57.5	84	60.4	204	66.4
Total cycles		40	100	139	100	307	100



(a) 8n13s



(b) 14n21s



(c) 11n23s

Fig. 3. The %Diff from the optimal solutions for various values of  $\alpha$

In our simulation, the traffic demands are randomly generated, and the ranges of the total working capacities for the three test networks are: 79 ~ 103 units in 8n13s, 101 ~ 140 units in 14n21s, and 80 ~ 102 units in 11n23s. The spare capacity for each working traffic is computed, and we use (%Diff) to represent the redundancy difference from the optimal solution. The simulation results for the three test networks are shown in Fig. 3. Fig 3(a) shows the same results

when  $\alpha$  is 0.5, 0.7 and 0.9 in 8n13s. It illustrates that when  $\alpha$  is 0.5, the 15 (37.5% of total cycles) candidates are good cycles, and they can deliver the average 2.39% for %Diff. The SCS algorithm achieves the best performance in 14n21s network; even when  $\alpha$  is 0.3, the 42 (30.2% of total cycles) candidates can deliver the average 0.74% for %Diff. All the results are very close to the optimal values as shown in Fig 3(b). In the 11n23s network, the results deviate from the optimal solutions when  $\alpha$  is 0.3 or 0.5. The average %Diff values are 13.27% and 7.8%, respectively. However, the %Diff values are small when  $\alpha$  is 0.7 or 0.9, they are 2.15% and 1.61% respectively for these cases. The numbers of candidates for both are about half of the total cycles. We summarize the average %Diff verse  $\alpha$  for the three test networks in Table III.

TABLE III  
THE AVERAGE REDUNDANCY DIFFERENCE (%DIFF) TO THE OPTIMAL SOLUTION

		8n13s	14n21s	11n23s
$\alpha$	0.3	5.70	0.74	13.27
	0.5	2.39	0.42	7.8
	0.7	2.39	0.42	2.15
	0.9	2.39	0.31	1.61

### B. Performance comparison to optimal design

In this section, we evaluate the SCS algorithm in two real networks; the COST239 and EON as shown in Figs 4(a) and 4(b). We set the  $\alpha$  as 0.3 in COST239 and 0.5 in EON, thus the numbers of candidate cycles are 954 (27% of the total) and 2886 (32.58% of the total), respectively.

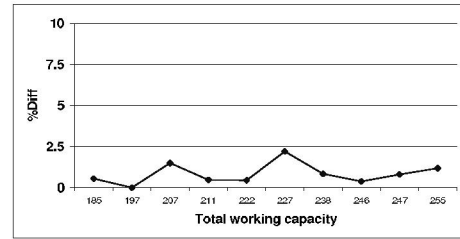
The simulation results are shown in Figs 4. From our computation, the average %Diff is 0.87% in COST239 (42.04% - 41.17%), and 3.7% in EON (66.06% - 62.36%). These results show that the SCS algorithm is able to generate small numbers of efficient  $p$ -cycles to minimize the spare capacity, and achieve 100% restorability within low computational complexity.

## V. CONCLUSION

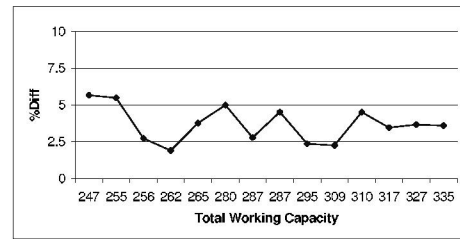
In this paper we have proposed the SCS algorithm for generating an efficient and sufficient set of  $p$ -cycles to achieve 100% working capacity protection, whilst also minimizing the spare capacity and reducing the time complexity. The simulation results have shown that the proposed algorithm can achieve optimal solutions for medium size networks, and close to optimal solutions for dense networks. In addition, the time complexity is greatly reduced for dense networks. Thus, we can conclude that the SCS algorithm balances the computational time and the optimality of solutions, and can be applied to large networks.

## REFERENCES

- [1] S. Ramamurthy, L. Sahasrabudde, and B. Mukherjee, "Survivable WDM mesh network," *Journal of Lightwave Technology*, vol. 21, no. 4, pp. 870–883, 2003.
- [2] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed pre-configuration: Ringlike speed with mesh-like capacity for self-planning network restoration," *Proc. IEEE International Conference on Communications (ICC) 98, Atlanta, Georgia, USA*, pp. 537–543, 1998.
- [3] W. D. Grover and D. Stamatelakis, "Self-organizing closed path configuration of restoration capacity in broadband mesh transport networks," *slides presented at Second Canadian Conference on Broadband Research (CCBR 98), Ottawa, Ontario, Canada, 1998*.
- [4] W. D. Grover and D. Stamatelakis, "Bridging thenring-mesh dichotomy with  $p$ -cycles," *Proceedings of Design of Reliable Communication Networks (DRCN 2000)*, 2000.
- [5] F. J. Blouin, A. Sack, W. D. Grover, and H. Nasrallah, "Benefits of  $p$ -cycles in a mixed protection and restoration approach," *Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), Banff, Alberta, Canada*, pp. 203–211, 2003.
- [6] W. D. Grover and J. Doucette, "Advances in optical network design with  $p$ -cycles: Joint optimization and pre-selection of candidate  $p$ -cycles," *Proc. IEEE LEOS Summer Topicals 2002, Mont Tremblant, Quebec, Canada*, pp. 49–50, 2002.
- [7] J. Doucette, D. He, W. D. Grover, and O. Yang, "Algorithmic approaches for efficient enumeration of candidate  $p$ -cycles and capacitated  $p$ -cycle network design," *Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003)*, pp. 212–220, 2003.
- [8] K. Lo, D. Habibi, Q. V. Phung, and H. N. Nguyen, "Dynamic  $p$ -cycles selection in optical WDM mesh networks," *IEEE Malaysia International Conference on Communications and IEEE International Conference on Networks, MICC-ICON 2005, 2005*.
- [9] H. Zhang and O. Yang, "Finding protection cycles in DWDM networks," *Proc. IEEE International Conference on Communications (ICC)*, pp. 2756–2760, 2002.
- [10] H. Zhang, O. Yang, JingWu, J. M. Savoie, T. Shan, and G. Wang, "A cycle-based rerouting scheme for wavelength-routed WDM networks," *Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW04)*, pp. 427–433, 2004.
- [11] C. Liu and L. Ruan, "Finding good candidate cycles for efficient  $p$ -cycle network design," *Proc. 13th International Conference on Computer Communications and Networks (ICCCN 2004)*, pp. 321–326, 2004.



(a) COST239



(b) EON

Fig. 4. Simulation results for the COST239 & EON Networks