2006

Efficient p-Cycle Design by Heuristic p-Cycle Selection and Refinement for Servivable WDM Mesh Networks

Kungmang Lo
Edith Cowan University

Daryoush Habibi
Edith Cowan University

Quoc Viet Phung
Edith Cowan University

Alexander Rassau
Edith Cowan University

Hoang N. Nguyen
Edith Cowan University

10.1109/GLOCOM.2006.429

Efficient p-Cycles Design By Heuristic p-Cycle Selection and Refinement for Survivable WDM Mesh Networks

Kungmeng Lo, Daryoush Habibi, *Senior Member, IEEE*, Quoc V. Phung, Alexander Rassau, and Hoang N. Nguyen
COMMUNICATIONS RESEARCH GROUP, SCHOOL OF ENGINEERING AND MATHEMATICS
EDITH COWAN UNIVERSITY
PERTH, AUSTRALIA

***Abstract*— Using p-Cycles to protect against single span failures in Wavelength-Division Multiplexing (WDM) networks has been widely studied. *p*-Cycle retains not only the speed of ring-like restoration, but also achieves the capacity efficiency over mesh networks. However, in selecting an optimal set of *p*-cycles to achieve the minimum spare capacity and fast computational time is an NP-hard problem. To address this issue, we propose a heuristic approach to iteratively select and refine a set of *p*-cycles, which contains two algorithms: the Heuristic *p*-Cycle Selection (HPS) algorithm, and the Refine Selected Cycles (RSC) algorithm. Our simulation results show that the proposed approach is within 3.5% redundancy difference from the optimal solution with very fast computation time even for large networks.**

I. INTRODUCTION

Survivability in backbone networks is a critical issue because network failure may cause huge data losses, especially in optical networks. Network survivability is to restore failed traffic flows via alternative backup routes [1]. The concept of p-cycle for span protection was proposed in 1998 [2] [3]. The nature of p-cycle is ring arrangement, hence p-cycle shows the same protection behaviors like a ring. Moreover, it can be configured efficiently over a mesh network [4], because the additional protecting capacities are provided to straddling spans without requiring any spare capacity. A straddling span is a span whose end-nodes belong to the p-cycle but the span itself is not part of the cycle.

The objectives of p-cycles are to minimize the spare capacity and to provide 100% restorability for the given traffic pattern. Two approaches of optimization have been studied and developed in the literature. One is to optimize the working capacity and the spare capacity simultaneously. This approach is known as the joint optimization [5] [6]. This sort of optimization will minimize the total capacity. In the other approach, namely the non-joint optimization [7] [8] [9], the working capacity and the spare capacity are minimized one after the other. This approach can reduce the computational complexity but the capacity utilization is not as efficient as the former.

The real protection efficiency of p-cycles in a network depends on lots of factors, e.g., the traffic pattern, network configuration, *etc*. In [5], the authors explain that a p-cycle with high *aprior efficiency (AE)* score has a high potential protection capability. The Integer Linear Program (ILP) is the well-known approach to p-cycles design. In order to achieve the optimal set of p-cycles, all cycles in the network should be taken into account. This is however impractical in large scale or dense networks because the number of candidates is too large. Limit the number of cycles candidates would reduce the complexity of the ILP models, but can only deliver the near optimal solutions. In this sense, selecting an efficient or *high merit* set of candidates becomes crucial and has been addressed in [6] [9] [8]. Another approach does not couple with the ILP model; but, it iteratively selects cycles based on a criteria of cycle selection, e.g., CIDA [7] and the ER-based unity-p-cycles design algorithm [8]. This approach can quickly produce the results, but the resulting redundancy has more than 5% deviation from the optimal value (redundancy is the ratio of the total spare capacity to the total working capacity).

In this paper we develop a heuristic p-cycle selection approach to solve this problem. In order to minimize the spare capacity and achieve 100% restorability for the working traffic, we first minimize the working capacity for the given traffic by an ILP model. In addition, all traffic is adaptively routed to avoid the shortage of capacity for *p*-cycles. Then, two heuristic algorithms are designed to determine a set of p-cycles which satisfy the design objectives. The results show that our approach delivers within 3.5% of redundancy difference from the optimal solution, with very fast computation time. The rest of this paper is organized as follows. In section II we review the related studies in the literature. We present our heuristic algorithms in section III. In section IV we evaluate the designed algorithms over two test topologies. Finally, we conclude our discussion in section V.

II. LITERATURE REVIEW

The first heuristic approach for selecting *p*-cycles is proposed by Doucette *et al.* [7]. In [7], two main tasks are performed: constructing cycles as candidates, and then determining a set of efficient cycles as protection channels. For constructing the candidate cycles, they first generate a set of primary cycles using the Stradling Link Algorithm (SLA) [10],

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

and then large cycles are formed by their 'Add', 'Join', or 'Grow' algorithms. The designed p-cycles are determined by iterative placement of the highest actual efficiency E_w. The $E_w(p)$ of a cycle p is computed as:

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

where w_l is the amount of unprotected working capacity on span l. $S_{p,l}$ is the number of protection channels related to l from the cycle p. $S_{p,l} = 1$ if l is an on-cycle span, and $S_{p,l} = 2$ if l is a straddling span. $\mathbb{L}$ is the set of spans in the network, and $cost_l$ is the cost of a unit capacity on l. This algorithm is known as the Capacitated Iterative Design Algorithm (CIDA). The best performance is achieved by CIDA-*Grow* algorithm, but the redundancy is 10% higher than the optimal value.

Another heuristic p-cycles selection in the literature is the *ER-based unity-p-cycle design algorithm*, proposed by Zhang *et al.* [8]. 'ER' is the efficiency ratio. The cycle selection here is very similar to the CIDA, but they consider the unidirectional traffic and unidirectional p-cycles in networks. Thus, they name a p-cycle as a unity-p-cycle, and all candidate cycles are considered in their model. 'ER' is the ratio of the number of actual-protected working units to the number of spare units of a unity-p-cycle. Given the traffic patterns (by centralized and distributed pattern) and network topology, the unity-p-cycles are iteratively selected with the maximum ER until 100% protection is achieved. Their results show 5.3% and 7.8% of redundancy difference from the optimal value for centralized and distributed traffic patterns, respectively.

III. p-Cycles Protection Design

In this section, we present our p-cycles protection design algorithm. For a given traffic, the first task is to minimize the working capacity, and preserve sufficient spare capacity for p-cycles. The second task is to minimize the spare capacity for those p-cycles, which can fully protect the working capacity. Our research is carried under the following assumptions:

- The network topology is bi-connected and undirected in which each fiber span is bidirectional.
- Each node has the same array of transmitters and receivers.
- Call requests are end-to-end connections.
- Full wavelength conversion is available at all nodes of the network.

A network topology can be modeled as a connected graph $G(N, L)$, where N and L are the number of nodes and spans in a network. Further definitions for the model are given below.

$\mathbb{L}$	the set of spans in a network, $\mathbb{L} = \{l_1, l_2 \ldots, l_L\}$.
$\mathbb{C}$	the set of capacity in a network, $\mathbb{C} = \{c_1, c_2, \ldots, c_L\}$.
$\mathbb{W}$	the set of (un-protected) working capacity in a network, $\mathbb{W} = \{\omega_1, \omega_2, \ldots, \omega_L\}$.
$\mathbb{U}$	the set of utilized capacity, $\mathbb{U} = \{u_1, u_2, \ldots, u_L\}$

Parameters:

c_l	the total capacity (channels) on the span l.
$\omega_{l(nd)}$	the number of working capacity on the span l, which incidents on a nodal degree (nd).
D	the number of given traffic.
W	the total working capacity.
$\mathbb{T}$	the set of given traffic in which t_d is the volume of demands d, $d \in \{1 \ldots D\}$.
t_d	the volume of traffic request for a connection d.
d	the connection route from node i to node j.
$p_{d,k}$	the k^{th} candidate route of connection d.
$\beta_{d,k}$	a indicator variable for candidate route $p_{d,k}$.
$\mathbb{P}$	the array of p-cycles in a network. $\mathbb{P} = \{cycle^i\}$, $i \in \{1 \ldots P\}$.
P	the total number of cycles in $\mathbb{P}$.
$cycle^i$	the i^{th} candidate p-cycle in $\mathbb{P}$, $i \in \{1 \ldots P\}$
	two expressions for $cycle^i$, they are $cycles_s$ and $cycles_p$.
$cycles_s$	the array of $cycle_{s,i}$ in $\mathbb{P}$, where $i \in \{1 \ldots P\}$.
$cycle_{s,i}$	the array of spare capacity required by $cycle^i$, and is expressed as $cycle_{s,i} = \{\mu_{i,1}, \mu_{i,2}, \ldots, \mu_{i,L}\}$.
$\mu_{i,l}$	the spare capacity of $cycle^i$ uses on span l. $\mu_{i,l} \in \{0, 1\}$, $\mu_{i,l} = 1$ if $cycle^i$ uses span l, $\mu_{i,l} = 0$ otherwise.
$cycles_p$	the array of $cycle_{p,i}$ in $\mathbb{P}$, where $i \in \{1 \ldots P\}$.
$cycle_{p,i}$	the array of protect-potential capacity by $cycle^i$, and is expressed as $cycle_{p,i} = \{\eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,L}\}$.
$\eta_{i,l}$	the protect-potential capacity of $cycle^i$ on span l. $\eta_{i,l} \in \{0, 1, 2\}$, $\eta_{i,l} = 2$ if $cycle^i$ is a straddling span on l, $\eta_{i,l} = 1$, if $cycle^i$ is a on-cycle span on l, and $\eta_{i,l} = 0$ otherwise.
$Cycle_s$	the array of selected cycles from the $cycle_{s,i}$.
$Cycle_p$	the array of selected cycles from the $cycle_{p,i}$.

A. Minimizing the Working Capacity

In a network, if traffic is known, the aim is to minimize the working capacity, and preserves sufficient spare capacity for p-cycles. This is done by an ILP model with appropriate constraints to satisfy these requirements. The designed ILP model is given below.

objective:

$$\textit{minimize} \quad W = \sum_{l=1}^{L} \omega_l \quad (2)$$

subject to:

$$\sum_{k=1}^{K} \beta_{d,k} = t_d \quad (3)$$

$$\omega_l = \sum_{d=1}^{D} \sum_{k=1}^{K} b_{d,k}^l \times \beta_{d,k}, \quad \forall l \in \mathbb{L} \quad (4)$$

where $b_{d,k}^l = 1$, if $p_{k,d}$ uses the span l; $b_{d,k}^l = 0$, otherwise. In addition, the routing paths of working traffic are constrained by the topology condition (nodal degree) as given by Eqns. (5) and (6).

$$\omega_{l(d=2)} \leq \frac{c_l}{2}, \quad \forall l \in \mathbb{L} \quad (5)$$

$$\omega_{l(d>2)} < c_l, \quad \forall l \in \mathbb{L} \quad (6)$$

B. Constructing all p-Cycles

The method for constructing the set of all p-cycles, P, is as follows: First, find a set of K shortest paths between the two end-nodes on each span (ie. span itself, when $K = 1$) in a network. Next, join any two paths to form a cycle if these

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

two paths are node-disjoint, so that the span of a cycle can be either straddling or on-cycle.

This method is the modified version of the Straddling Link Algorithm (SLA) [8] [10]. Please note that all cycles are generated by increasing the number of K until the total number of cycles cannot be increased. Thus, we need to trial the minimum value of K to find the total number of cycles. This may take some computation time. However, determining all cycles in a network is a one-off task, because such information can be stored and recalled if required. The purpose for considering all p-cycles as candidates is to avoid losing good cycles.

C. p-Cycles Selection Design

The p-cycle selection is completed using the *Heuristic p-Cycle Selection* (HPS) algorithm and the *Refine Selected Cycles* (RSC) algorithm. The first algorithm iteratively selects a cycle with the best *Cycle Efficiency* (CE) until full protection is achieved, and the second algorithm refines the selected cycles. Details are given below.

1) Heuristic p-Cycle Selection (HPS) Algorithm: Firstly, a qualified candidate *p*-cycle must satisfy the span capacity constraint as shown in Eqn. (7). If not, the cycle will be removed in the next step.

$$\mathbb{U} + cycle_{s,i} \leq \mathbb{C}, \quad i = \{1, 2, \ldots, P\} \tag{7}$$

Secondly, CE is calculated for all qualified candidates. The CE consists of three factors: the cycle weight ϖ, the waste capacity Γ, and the effective straddling spans ϵ:

- ϖ is the ratio of the total actual protected capacity $(\sum ap)$ with the power n (the control parameter) to the total spare capacity required $(\sum \mu)$. ϖ_i for the $cycle^i$ is expressed as
$$\varpi_i = \frac{(\sum_{l=1}^{L} ap_{i,l})^n}{\sum_{l=1}^{L} \mu_{i,l}}, \quad i = \{1, 2, \ldots, P\} \tag{8}$$
where n is a real number, and is designed to control the weight of protected capacity to the spare capacity, ie. if $n = 1$, both capacities have the same weight, and ϖ is equal to Eqn. (1). If $n > 1$, then the actual protection capability has a higher weight than the spare capacity and vice versa. $ap_{i,l}$ is the actual protected capacity by $cycle^i$ at span l; $ap_{i,l} \in \{0, 1, \eta_{i,l}\}$; $ap_{i,l} = \eta_{i,l}$, if $\omega_l \geq 2$; $ap_{i,l} = 1$, if $\omega_l = 1$, and $\eta_{i,l} \neq 0$, $ap_{i,l} = 0$, otherwise. $\mu_{i,l} \in cycle_{s,l}$.
Another array AP_i, which is a span array to indicate the actual protection capacity $ap_{i,l}$ at every span l by $cycle^i$, is expressed as
$$AP_i = \{ap_{i,1}, ap_{i,2}, \ldots ap_{i,L}\}, \quad i = \{1, 2, \ldots, P\} \tag{9}$$
- Γ is defined as the idle capacity on the on-cycle spans. If an on-cycle span does not protect the working capacity, we consider this capacity as wasted because an on-cycle span may take one spare capacity. Therefore a cycle with a lower value of Γ has better capacity utilization. Γ_i of $cycle^i$ is computed as
$$\Gamma_i = \sum_{l=1}^{L} \gamma_l, \quad i = \{1, 2, \ldots, P\} \tag{10}$$
where $\gamma_l = 1$, if $\mu_{i,l} > \omega_l$; $\gamma_l = 0$, otherswise.
- ϵ is defined the number of the actual protected straddling capacity. A straddling span of a cycle can restore two units of working capacity without requiring any spare capacity. Thus, more straddling spans means less spare capacity required. ϵ_i for $cycle^i$ is calculated by
$$\epsilon_i = \sum_{l=1}^{L} sd_l, \quad i = \{1, 2, \ldots, P\} \tag{11}$$
where
$$sd_l = \begin{cases} \min(\omega_l, \eta_{i,l}), & \text{if } \eta_{i,l} = 2,\ \eta_{i,l} \in cycle_{p,l} \\ 0, & \text{otherwise} \end{cases}, \quad \forall l \in \mathbb{L}$$

The cycle with the best CE is then selected and stored in two forms: $Cycle_s$ and $Cycle_p$ for further calculation. Thirdly, update the un-protected working capacity $\mathbb{W}$ and the utilized capacity $\mathbb{U}$. Finally, repeat the previous steps until the total unprotected working capacity is zero. The pseudocode of the HPS algorithm is given in Algorithm 1.

Algorithm 1 : Heuristic *p*-Cycle Selection (HPS)

Require: $\mathbb{P}(cycles_s, cycles_p), \mathbb{C}, \mathbb{W}$
Ensure: $Cycle_s, Cycle_P$
  $Cycle_l \leftarrow \phi, \quad Cycle_P \leftarrow \phi, \quad \mathbb{U} = \mathbb{W}$
  while $\sum |\mathbb{W}| > 0$ **do**
    $cycle^i(cycle_{s,i}, cycle_{p,i}) \leftarrow \mathbb{P}(cycles_s, cycles_p)$
    $(\varpi, \Gamma, \epsilon, AP) \leftarrow \phi$
    for $i = 1$ to P **do**
      if $cycle^i$ satisfies Eqn. (7) **then**
        Do $(\varpi_i, \Gamma_i, \epsilon_i, AP_i)$ by Eqns. (8), (9), (10), and (11)
        if $\varpi_i > \varpi$ **then**
          $(\varpi, \Gamma, \epsilon, AP) \leftarrow (\varpi_i, \Gamma_i, \epsilon_i, AP_i)$
        else if $\varpi_i = \varpi$ **then**
          if $\Gamma_i \leq \Gamma$ or $\epsilon_i \geq \epsilon$ **then**
            $(\varpi, \Gamma, \epsilon, AP) \leftarrow (\varpi_i, \Gamma_i, \epsilon_i, AP_i)$
          end if
        end if
      end if
    end for
    $Cycle_s \leftarrow cycle_{s,i}$, $Cycle_p \leftarrow cycle_{p,i}$,
    $\mathbb{U} = \mathbb{U} + cycle_s^i$, $\mathbb{W} = |\mathbb{W} - AP|$
  end while

2) Refine Selected Cycles (RSC) Algorithm: The purpose of RSC is to minimize the redundancy by refining the cycles selected by the HPS algorithm. The motivation to design RSC is that HPS iteratively selects a cycle with the best CE, which only counts on the actual protection capability of the cycle itself, and neglects the unprotected traffic pattern in the network. Therefore, the spare capacity has room for

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

improvement. The method to refine the selected cycles is to search for any two or more cycles that can be replaced by other cycles which require less spare capacity and still protect all traffic. In our study, RSC can efficiently remove some wasted spare capacity. However, when it searches for more complex replacements, i.e. finding more cycles to split and join simultaneously, the computation time will be a little higher. For this reason, RSC is set to merge two cycles to become one cycle. The pseudocode of the RSC algorithm is given in Algorithm 2.

Algorithm 2 : Refine Selected Cycles (RSC)

Require: $\mathbb{P}, Cycle_s$, the original $\mathbb{W}$
Ensure: Set of refined cycles P_r

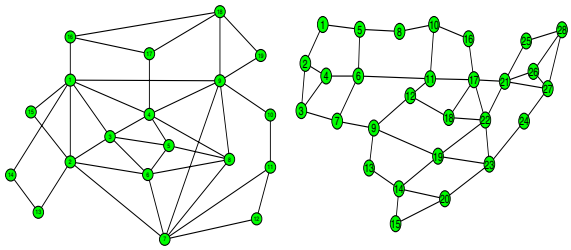
```
isRefined ← true
while (isRefined) do
  flag ← 0
  for each pair of (cycle^i, cycle^j) ∈ Cycle_s do
    P_tmp ← ℙ \ {cycle^i, cycle^j}
    for each cycle cycle^k ∈ ℙ do
      P_tmp1 ← P_tmp ∪ cycle^k
      if P_tmp1 can protect 𝕎 then
        flag ← 1
        Break
      end if
    end for
    if flag == 1 then
      Break;
    end if
  end for
  if flag == 0 then
    isRefined ← false
  else
    Cycle_s ← P_tmp1
  end if
end while
P_r ← Cycle_s
```

IV. Simulation Results and Discussion

We evaluate and implement the proposed algorithms into two test networks, the EON and USA [7] networks as shown in Fig 1. Our computing platform is an IBM ThinkCentre PC with an Intel Pentium IV 3.0-GHz processor, with 1 GB of RAM, running Windows XP. We assume that the cost of a p-cycle is the number of hops and each span contains 24 channels ($C = 24$). The total number of p-cycles, the average AE values and the average hop length in the EON and USA networks are given in Table I.

A. Analyzing the Value of the Control Parameter n

We first observe the performance of the HPS algorithm, when the values of n is taken to be 0.5, 1, 1.5, 2 ,2.5, 3, 5 and 10 in Eqn. (8). The given traffic D is randomly generated. The working capacity pattern (distribution) $\mathbb{W}$ and the total working capacity W are calculated by Eqns. (2) $\sim$ (6). In the EON network, D is from 85 to 170; W is from 176 to 335 units, and the average is 277 units. In the USA network, D is from 95 to 180, W is from 219 to 377 units, and the average W is 298 units. The variation of the average capacity from the optimal value and the computation speeds are presented in Table II.



(a) EON (N = 19, L = 38, $\bar{d} = 4$) (b) USA (N = 28, L = 45, $\bar{d} = 3.214$) [7]

Fig. 1. The topologies of the EON & USA networks

TABLE I
p-Cycles in the EON & USA Networks

	EON	USA
P	8857	7321
Average AE value	2.57	1.70
Average hop number	11.38	18.65

TABLE II
Performance Comparison in the Test Networks

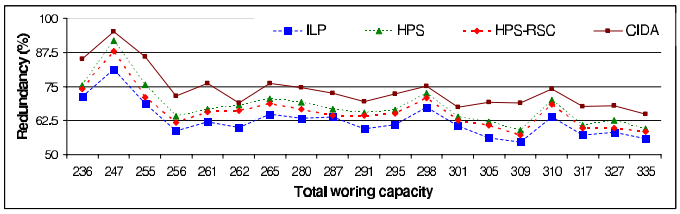
$n =$	0.5	1	1.5	2	2.5	3	5	10
	The variation of the average spare capacity from the optimal (units)							
EON	107	20.55	7	6.8	6.6	7.25	8.25	8.25
USA	137.1	16.71	9.79	7.92	7.79	7.5	7.54	7.54
	The average computational time (second)							
EON	13.39	3.92	2.55	2.19	2.21	2.16	2.17	2.24
USA	14.37	2.94	1.79	1.58	1.57	1.50	1.50	1.55

From the numerical results, we can see that the control parameter n influences the cycle selection and the overall performance. In general, when $n = 2.5 \sim 3$, the spare capacity deviation from the optimal value is small and the computation time is short. The worst case is when $n = 0.5$, which results in capacity deviation and takes longer computation. In addition, when $n = 1$, the results are worse than when $n > 1$. Thus, the value $2.5 \sim 3$ for n is best in the HPS algorithm. From these results we can also predict that the performance will be better in [7] if the actual efficiency $E_w(p)$ in Eqn. (1) is modified with the power $n = 2.5 \sim 3$. In the next section, we select $n = 2.5$ for the EON network, and $n = 3$ for the USA network.

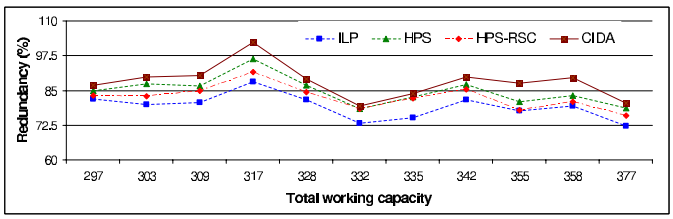
B. Performance Comparison

In this section, we compare the performance for the ILP model, the HPS algorithm, the HPS-RSC algorithm and the CIDA algorithm [7] in the test networks. Note that the CIDA

1-4244-0357-X/06/$20.00 ©2006 IEEE



(a) EON



(b) USA

Fig. 2. The redundancy results in EON & USA networks

algorithm is given the complete set of cycles P in our simulation.

The total working capacity, W, in the EON and USA networks are taken to be from 236 to 335 units, and from 297 to 377 units, respectively. The simulation results are shown in Fig. 2, and in Table III. In general, the computation time for the ILP model is much longer because of the large number of candidates. The average speed for the ILP model in USA is faster than in EON. Although the size of the USA network is larger than EON, it is not a dense network ($\bar{d} = 3.2$, 7 nodes, nodal degree 2). Note that the saw-tooth shape of the redundancy distribution is caused by the traffic pattern and the corresponding p-cycle matching in the network topology, but it is not related to the scale of the traffic capacity.

TABLE III
SIMULATION RESULTS IN THE EON & USA NETWORKS

		Redundancy (%)			Speed (sec)	
		average	s.d.	%Diff	average	s.d.
EON ($n = 2.5$)	ILP	62.51	6.40	0	3793.09	-
	HPS	67.91	7.59	5.4	1.81	0.25
	HPS-RSC	65.94	7.01	3.4	3.04	0.47
	CIDA	73.86	7.58	11.35	2.63	0.43
USA ($n = 3.0$)	ILP	79.20	4.50	0	831.31	-
	HPS	84.83	5.02	5.36	1.38	0.16
	HPS-RSC	82.49	4.27	3.29	2.6	0.48
	CIDA	88.09	6.05	8.89	2.11	0.28

From the numerical results, the HPS-RSC algorithm delivers a lower redundancy than the HPS and CIDA. In addition, it achieves $3\% \sim 3.5\%$ of redundancy difference from the optimal value.

These results can be explained from two perspectives. Firstly, the Cycle Efficiency (CE) in HPS algorithm can directly select the *high merit* p-cycles, so the redundancy performance is lower than CIDA. Secondly, most of the selected p-cycles is then re-constructed by the RSC algorithm, thus the spare capacity usage is reduced. Overall, the RSC can save about 2% of redundancy in the test networks. Although the average computation time of the HPS-RSC could be longer than the HPS and CIDA algorithms, it is still acceptable.

V. CONCLUSION

In this paper, we have presented an efficient heuristic approach for p-cycles design in WDM mesh networks. The working capacity is minimized by the designed ILP model and the p-cycles are selected using the HPS-RSC algorithm. The proposed *Cycle Efficiency* and *Refine Cycles* in HPS-RSC can efficiently select and construct the protection cycles, so that HPS-RSC achieves less than 3.5% redundancy difference from the optimal value, with about 3 seconds computation time in large networks. Thus, we can conclude that the HPS-RSC algorithm delivers close to the optimal solutions with low computation complexity. Moreover, we can predict that the HPS-RSC algorithm can be applied in dynamic traffic environments.

REFERENCES

[1] S. Ramamurthy, L. Sahasrabuddhe, 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] 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.

[5] 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, Qubec, Canada,*, pp. 49–50, 2002.

[6] D. Schupke, C. Gruber, and A.Autenrieth, "Optimal configuration of p-cycles in WDM networks," *IEEE*, pp. 2761–2765, 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] 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.

[9] 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.

[10] H. Zhang and O. Yang, "Finding protection cycles in DWDM networks," *Proc. IEEE International Conference on Communications (ICC)*, pp. 2756–2760, 2002.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.