

# Multi-Criteria p-Cycle Network Design

Hamza Drid, Bernard Cousin, Samer Lahoud  
University of Rennes I -IRISA-  
Rennes, France  
Email: {hdrid, bcousin, slahoud}@irisa.fr

Miklos Molnar  
INSA of Rennes -IRISA-  
Rennes, France  
E-mail : molnar@irisa.fr

**Abstract**—The major challenge of p-cycle network design resides in finding an optimal set of p-cycles protecting the network for a given working capacity. Existing solutions (exact and heuristic approaches), for solving the problem, find the set of p-cycles protecting the network through two steps: one step for generating candidate p-cycles and a second step for selecting the efficient ones. In this paper, we present a novel heuristic approach, which computes an efficient set of p-cycles protecting the network in one step. Our heuristic approach takes into consideration two different criteria: the redundancy and the number of p-cycles involved in the solution. Simulation study shows that our approach necessitates a lower redundancy and fewer p-cycles to protect the network compared to state-of-the-art approaches.

**Index Terms**—WDM protection, Network survivability, p-cycle, Network management complexity.

## I. INTRODUCTION

Optical WDM mesh networks are able to transport huge amounts of information. The use of such technology, however, poses the problem of protection against failures such as fiber cuts or node failures. Consequently, any cut of such fiber may lead to huge data loss and a lot of traffic being blocked. For this reason, methods of protection should be implemented to minimize the data loss when a failure occurs. One of the principal protection methods proposed for optical WDM networks is based on p-cycles or preconfigured protection cycles, introduced by Grover and Stamatelakis in [1].

P-cycle offers the advantages of both ring<sup>1</sup> and mesh protection schemes: faster restoration time as in ring protection, and high capacity efficiency as in mesh-protection. Precisely, in p-cycle, restoration time is faster because only the two end nodes of the failed link need to perform restoration. Moreover, capacity efficiency is due to the fact that a p-cycle can provide protection not only for on-cycle links but also for straddling links. A straddling link is a link, which does not belong to the p-cycle but whose end-nodes are both on the p-cycle. In addition, p-cycle offer two restoration paths to the failed straddling links without requiring any additional spare capacity. This property reduces effectively the required protection capacities.

Figure 1 depicts an example that illustrates p-cycle protection. In figure 1(a), (a-b-c-d-e-a) is a p-cycle with one unit of spare capacity on each on-cycle link. When the on-cycle

link (d-c) fails as show in figure 1(b), the p-cycle provides one protection path (d-e-a-b-c). In figure 1(c) an example of a straddling link failure is shown. When the straddling link (e-c) breaks, the p-cycle can protect two working wavelengths on this link by providing respectively two alternate paths (e-d-c) and (e-a-b-c).

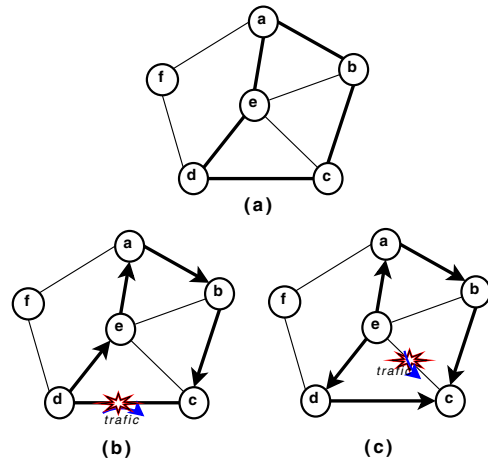


Fig. 1. Protection using p-Cycle

A p-cycle with high efficiency is a p-cycle that has a small ratio between its spare and protected capacity, i.e. it protects more working wavelengths using few spare wavelengths.

The major challenge of this method of protection resides in finding the optimal set of p-cycles in terms of resource utilization that protects the network for a given working capacity distribution. The p-cycle design can be formulated either as a non-joint or a joint optimization problem. In the first approach, after the working paths are routed (e.g. using shortest paths) the optimal set of p-cycles is calculated using available capacity [1], [2], [3], [4], [5]. In the second approach, the routing of the working paths and the p-cycles are computed simultaneously optimizing the total capacity [6], [7], [10]. Solving the joint optimization problem is more difficult because of the additional computation complexity.

Several solutions have been proposed in the literature to solve these optimization problems. These solutions can be classified into two classes: exact and heuristic solutions. The first class generally uses Integer Linear Programming (ILP) to find the optimal solution. However, ILP methods becomes unsuitable as the size of the network increases, because the

<sup>1</sup>In the ring protection, the traffic on the failed link or node is rerouted around the ring on the protection fibers between the nodes adjacent to the failure.

number of p-cycles in a graph grows exponentially with the network size. The second class is heuristic solutions, divided into two sub-classes: heuristic approaches based on ILP formulation and pure heuristic approaches. In the first sub-class, a limited set of candidate p-cycles is generated, and then the efficient sub-set of p-cycles is selected using ILP formulation [1]. In order to find a good solution with this approach it is necessary to generate a large number of candidate p-cycles. This means that the running time for solving the ILP formulation increases dramatically [8], [9]. The second sub-class (pure heuristic) tries to find a good solution without using ILP formulation. The objective of this kind of solutions is to reduce the time required to compute an efficient set of p-cycles that protects the network [3], [5].

In this paper, we focus on algorithms proposed for the non-joint protection problem. Existing solutions start by generating a set of candidate p-cycles, and then select a good sub-set of p-cycles. Most of the solutions proposed in the literature consider the p-cycles which have more straddling links as the most efficient p-cycles. Consequently, the p-cycle generation is based only on the topology of the network and is completely independent of the working capacity distribution. Another drawback of these solutions is that they generate a large number of candidate p-cycles, which increases running time of the algorithm. In this paper we propose a new heuristic to compute a set of p-cycles that protects the network without going through the step of candidate p-cycle generation. Our heuristic is independent of ILP and takes into account the working capacity of network. These enable us to find rapidly the efficient set of p-cycle protecting the network.

The rest of the paper is organized as follows. In section 2, we describe our algorithm of p-cycle generation based on the incremental aggregation of cycles. In section 3 we evaluate our heuristic and compare it with the exact solution that uses ILP and with the main proposed heuristics. We conclude the paper in section 4.

## II. OUR HEURISTIC

### A. Network Model

We model the WDM optical network as an undirected graph  $G = (V, E)$  where each node in  $V$  represents an optical switch and each edge in  $E$  represents a network link. Each link  $j$  has  $w_j$  working wavelength channels. This working capacity is obtained by routing the traffic demands. According to the p-cycle definition a p-cycle is formed in the spare capacity of the network. A p-cycle can protect one wavelength on each one of its on-cycle link and two wavelengths on each of its straddling link using only one unit of wavelength on each of its on-cycle links. In this case the spare capacity of a p-cycle can be defined as the number of on-cycle links of the p-cycle. In this paper, we consider the most frequent failure in the optical networks, which is the single link failure. In this paper we consider the model that ensures full protection against single failures [11].

Figure 2(a) shows a WDM network with eight optical switches (a...h), where each link is associated with an integer

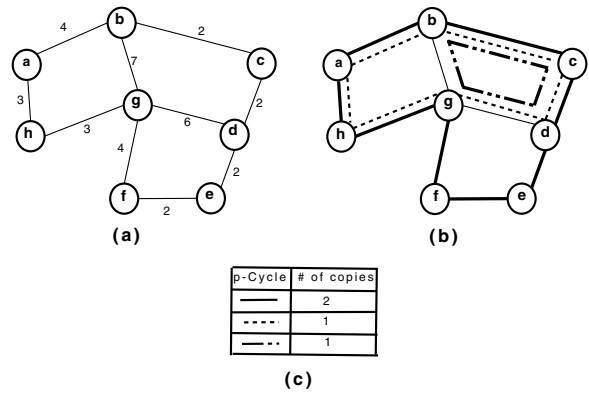


Fig. 2. P-cycle example

value that indicates the working capacity of the link. Figure 2(b) shows a set of p-cycles that protect the network for the given working capacity. The table shown in figure 2(c) contains a set of p-cycle structures protecting the network, where "# copies" denotes the number of copies of each p-cycle structure. For instance, the first p-cycle structure (a-b-c-d-e-f-g-h-a) in the table has two copies, each one can protect one wavelength on each link belonging to the p-cycle and two wavelengths on the straddling links (b-g) and (d-g).

### B. Algorithm Motivation

In this section, we describe the motivation of our algorithmic approach for p-cycle design. We start by analyzing the limitations of the heuristics proposed in the literature. Then we introduce the main guidelines of our heuristic that enable to overcome these limitations.

Firstly, the proposed heuristics, pure or ILP-based, use a two-steps approach that starts by generating a set of candidate p-cycles and then selects the efficient ones for protection. Hence, the quality of the solution depends heavily on the set of candidate p-cycles. Particularly, these heuristics need to generate a larger set of candidate p-cycles in order to increase the search space and enhance the efficiency of the final solution. However, this leads to considerable increase in computation time and induces a serious limitation for these approaches. In our solution, we propose a one-step algorithm that directly computes the p-cycles that are used for protection. This enables to perform a fine tuning between efficiency and running time and overcomes the limitation of using a set of candidate p-cycles.

Secondly, the proposed heuristics generate the set of candidate p-cycles based only on the network topology. In order to find an efficient solution in terms of resource utilization (redundancy), they consider that p-cycles with a large number of straddling links are more efficient, hence they are preferably selected for protection. However, this may not be always advantageous: consider for example the case where the use of p-cycles with a large number of straddling links results in over-protecting some links and thus reduces the efficiency of the global solution as shown in figure 3. The p-cycle (a-b-

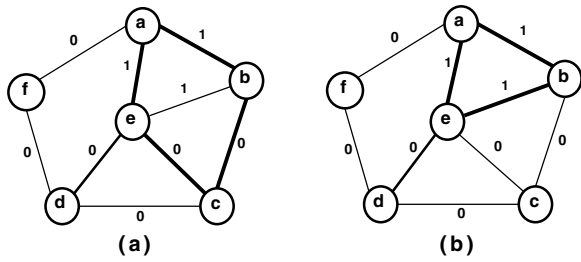


Fig. 3. Efficient p-cycle example

e-a) shown in figure 3(b) is more efficient than the p-cycle (a-b-c-e-a) shown in figure 3(a) in spite that the latter one contains a straddling link. In our heuristic, we overcome this drawback by taking into account the working capacity on the network links. Thus the number of straddling links per p-cycle is adapted to the working capacity in order to ensure better efficiency of the solution.

Thirdly, the objective of our heuristic is to find a good set of p-cycles not only in term of resource utilization (redundancy) like the other proposed heuristics. But it takes into consideration two different criteria: the redundancy and the number of p-cycles in the final set of protection. These criteria are defined in the following paragraph.

The redundancy criterion enables to measure the protection efficiency: it denotes the ratio between the protection resources and the working resources in the network. Redundancy constitutes a major evaluation criterion for WDM network design that gives an insight on the quality and the relative cost of network protection. In our approach, we strengthen this choice by integrating the redundancy in the iterative procedure of the algorithm.

In the following, the network redundancy is denoted by  $R$ . Particularly,  $R(c)$  denotes the redundancy of cycle  $c$  and is defined as the ratio between the spare capacity (number of wavelengths used by this cycle) and the working capacity of the cycle (number of wavelengths protected by the cycle on its on-cycle and straddling links). Similarly, the redundancy  $R$  of the final solution is computed as the ratio between the total network spare capacity used for protection design and the protected working capacity in the network. The number of p-cycles is another evaluation criterion for our solution. It is motivated by the fact that solutions with a small number of structures simplify the network management task [11].

For example, consider the case where the network designers manually implement the protection design. In this case, solutions with a small number of p-cycles alleviate the tedious configuration and maintenance tasks and reduce the risk of configuration errors. This approach remains valid when using an automated control plane such as the GMPLS architecture. In this case, a solution with a reduced number of protection structures enables to reduce the stress over the control plane and eases the management task. The details of our algorithm are presented in the following section. The main driver of our algorithm is to overcome the aforementioned

limitations that affect the state-of-the-art solutions. Thus, our algorithm computes an efficient set of p-cycles that protects the network and takes into account the working capacity and the topology of network. Our algorithm performs in a single step without undergoing a preliminary step for candidate p-cycle generation. Additionally, our algorithm explicitly takes into account two efficiency criteria: the redundancy and the number of p-cycles.

### C. Algorithm Insights

In this section we describe our algorithm for computing p-cycles. In a preliminary step of our algorithm, with the help of the algorithm introduced in [13], we start by computing the set of shortest cycles in the network. Then, we use this set of shortest cycles to construct the set of p-cycles (by incremental aggregations) that protects the network working capacity as explained in the following.

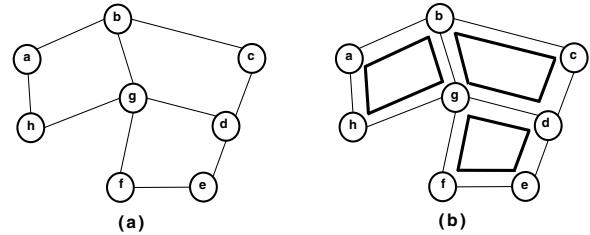


Fig. 4. Figure 4. Set of shortest cycles

Each iteration of our algorithm starts by selecting one shortest cycle among all the shortest cycles in the network. Typically, we choose the cycle containing a link with minimal non-zero working capacity (cf. Figure 5, step 4). Let us denote this current cycle by  $c'$ . Then we search for a cycle that is to be aggregated with cycle  $c'$ . The set of eligible cycles for aggregation with cycle  $c'$  must satisfy the following three conditions: first, an eligible cycle shares one and only one link with cycle  $c'$ , second an eligible cycle shares no nodes with cycle  $c'$  except the end nodes of the shared link (cf. Figure 5, step 5).

We choose to aggregate the eligible cycle that leads to the largest reduction in redundancy, if and only if the redundancy of the newly aggregated cycle is lowest than that of cycle  $c'$  (cf. Figure 5, step 6).

We now consider the new aggregated cycle as the cycle  $c'$  and proceed with another aggregation. The aggregation process eventually stops when the eligible set becomes empty or no aggregation leads to a reduction in the redundancy. The output aggregated cycle of one iteration is added to the final protection p-cycles set. The working capacities that are protected by the newly computed p-cycle are removed from the corresponding links and the algorithm iterates until all the working capacity is protected or there is no more available wavelengths in the network (cf. Figure 5, step 7 and 8).

Algorithm: P-cycle generation
<b>Input :</b> Set of shortest cycles, working capacity of each link
<b>Output:</b> Set of protection p-cycles
<p>(1) <math>\forall j \in E, u_j = w_j;</math></p> <p>(2) Let iteration number <math>i = 1;</math></p> <p>(3) <math>\forall j, k \in E,</math> select <math>j</math> such that <math>0 \leq u_j \leq u_k;</math></p> <p>(4) Select the shortest cycle <math>c'</math> contain <math>j.</math> (such that <math>c'</math> has the biggest number of unprotected links between all the shortest cycles using link <math>j</math>);</p> <p>(5) Select a set of shortest cycles <math>SSC = \{ c_k / k &gt; 0 \}</math> such that each shortest cycle verifies the following conditions:  a. Cycles <math>c'</math> and <math>c_k</math> share one and only one link;  b. Cycles <math>c'</math> and <math>c_k</math> do not share any node except the end nodes of the shared link;</p> <p>(6) If ( <math>SSC \neq \text{empty}</math> ) then  a. Select the cycle <math>c_k</math> that has the small value of <math>R(c'+c_k);</math>  b. If <math>(R(c'+c_k) \leq R(c'))</math> then <math>c' := c'+c_k</math> and go to step (5);</p> <p>(7) For each link <math>j</math> protected by the P-cycle update the value of <math>u_j</math> according to :  a. <math>u_j = u_j - 1</math> for each link <math>j</math> on cycle <math>c';</math>  b. <math>u_j = u_j - 2</math> for each straddling link of cycle <math>c';</math></p> <p>(8) If <math>j \in E</math> with <math>u_j &gt; 0</math> and if there is available capacity in the network  a. let <math>i = i + 1</math> and go to step (3);</p>
<p><math>w_j</math>: Working capacity of link <math>j.</math>  <math>u_j</math>: Unprotected working capacity of link <math>j.</math>  <math>(x+y)</math>: The aggregation of cycle <math>x</math> with the cycle <math>y.</math>  <math>R(c)</math>: The redundancy of cycle <math>c.</math></p>

Fig. 5. Algorithm description

### III. ALGORITHM EXTENSION

In this section, we introduce an extension of our algorithm that modifies the aggregation process. This extension enables to improve the overall performance as confirmed by the simulation results in section IV. The basic idea is to relax the redundancy constraint in the incremental aggregation process. Precisely, the basic version of our algorithm introduced in section 2 required that the redundancy must be strictly reduced when two cycles are aggregated. Thus, no cycle aggregation leading to an increase in redundancy is performed. However, this constraint leads to inefficiency particularly when several successive aggregations are needed to improve the overall redundancy. This is typically the case whenever a first aggregation is inefficient (increases the redundancy) but if performed, it may be followed by very efficient aggregations. On a wider scope, this is a well-known technique that is used in order to avoid local minima in the search for a global minimum.

In the extended algorithm, we introduce a new variable, that indicates if large p-cycles may turn to be efficient or not. This variable, called ULR, denotes the ratio between the number of unprotected links and the total number of links.

$$ULR = \frac{Nb\_Unprotected\_Links}{|E|} \quad (1)$$

$$ULR = \frac{|E| - Nb\_Protected\_Links}{|E|} \quad (2)$$

Where  $|E|$  is the number of links in the network,  $Nb\_Unprotected\_Links$  is the number of unprotected links in the network,  $Nb\_Protected\_Links$  is the number of protected links in the network.

Intuitively, when the number of unprotected links is relatively high, i.e. for larger values of URL, large p-cycles may improve the redundancy and become efficient and vice-versa. Therefore, the new condition for aggregating cycles tolerates an increase in the redundancy whenever the value of ULR is large. For this propose, the condition for aggregating cycle  $c$  and cycle  $c'$  (cf. figure 5, step 6) becomes:

$$R(c' + c_k) \text{ Or } x \leq ULR \quad (3)$$

When the aggregation does not reduce the redundancy (i.e. the first sub-condition of (3) is not verified), we generate randomly a real number  $x$  between 0 and 1 (uniform distribution). If the generated number is less than ULR, then we make the aggregation. Therefore, larger values of URL (larger ratio of unprotected links) lead to more aggregations and generate bigger cycles, whereas smaller values of URL lead to smaller cycles.

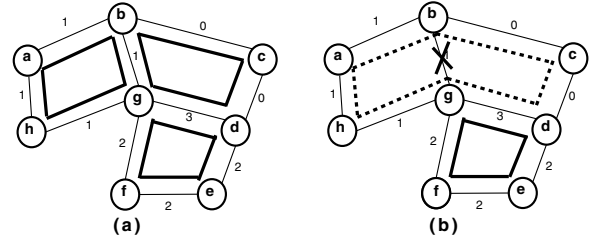


Fig. 6. P-cycles aggregation (Basic Solution)

Let us see how this extension performs on a small example. Figure 6(a) shows the state of the network at an intermediary iteration of the algorithm (some links have no working capacity to protect), and the set of shortest cycles (a-b-g-h-a, b-c-d-g-b, g-d-e-f-g) that we will use to construct p-cycles. At this stage, the algorithm selects the cycle (a-b-g-h-a) using the link with minimal non-zero working capacity as the  $c'$  cycle (cf. figure 5, step 3 and 4). The SSC set obtained after the selection of the  $c'$  cycle contains one cycle (b-c-d-g-b) (cf. figure 5, step 5). With the basic algorithm the aggregation of cycle  $c'$  and cycle (b-c-d-g-b) can not be performed because:  $R(a-b-c-d-g-h-a) = 6/4$  ;  $R(a-b-g-h-a) = 4/4$ .

However, it appears to be useful to perform this first aggregation (despite its inefficiency) because it leads to an efficient aggregation afterwards. With the new extension, as shown in figure 7 (b), the cycle  $c'$  and the cycle (b-c-d-g-b) are permitted to be aggregated with the probability of 0.8 i.e.  $ULR = 8/10$ . If the generated random number is between 0 and 0.8, then the aggregation is performed and the new  $c'$  cycle

becomes (a-b-c-d-g-h-a). The next aggregation of  $c'$  with its neighbor (g-d-e-f-g) leads to a global reduction in redundancy  $R(a-b-c-d-e-f-g-h-a)=8/9$  as show in figure 7 (c).

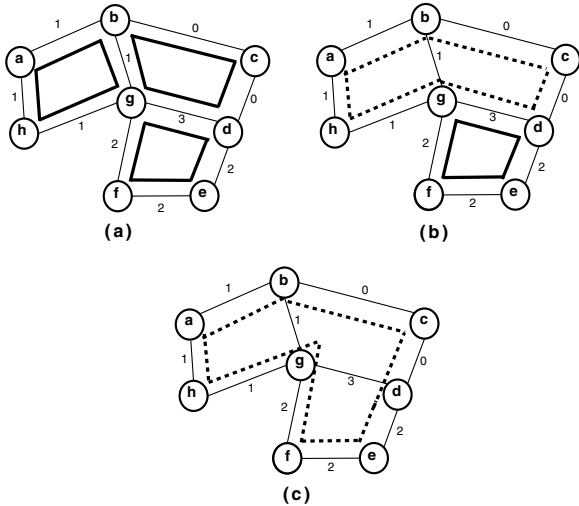


Fig. 7. P-cycles aggregation (Extended Solution)

At the end of iteration of the extended solution, the obtained cycle  $c'$  is not necessarily the cycle with the small redundancy, because in this version of algorithm, the aggregation is not conditioned by the reduction of the redundancy. For this propose, during the aggregation processes of this new version of algorithm we keep the cycle obtained after each aggregation, and at the end of the aggregation process we select the cycle having the smallest value of redundancy. The selected cycle is added in the final set of p-cycle protecting the network.

#### IV. SIMULATION RESULTS AND ANALYSIS

In this section, we evaluate our algorithm in terms of redundancy and in terms of number of p-cycle structures obtained in the final protection set. Simulation experiments are carried out using two widely used network topologies: the pan-European COST 239 topology and the KL topology. The first is an 11-node and 26-link network, taken from [11], the second is a 15-node and 28-link taken from [12]. COST239 has a high nodal degree ( $d=4.73$ ) which lead to a high number of distinct cycles (3531 cycles). KL network has 1600 distinct cycles, and it nodal degree is 3.73. The test networks are shown in figure 8.

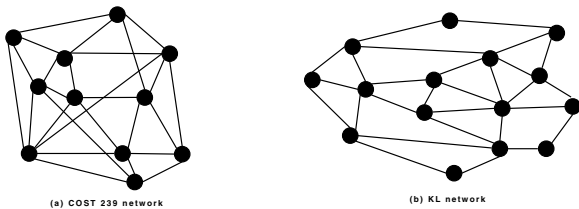


Fig. 8. Test Networks

The traffic demand is uniformly distributed among all source-destination pairs. For each source-destination pairs, an

TABLE I  
AVERAE TARFFIC

	COST 239	KL
Demands	276.7	524.9
Working capacity	422.2	1118.1

integer number between zero and the maximum allowable demand (which is 10 in our simulation) is randomly generated. The working capacity on every network link is calculated after all demands have been routed using the shortest-path routing. Ten experiments are performed for each topology network and the average values are presented as the final results.

In our simulation we assume that the capacity of the network was designed to accept all connection demands and to ensure 100% protection. Through the simulation we assume that each node in the network has full wavelength conversion capability. All simulations are running on DELL Quadri Dual Core Xeon processor and 4 GB of RAM. The machines run Windows 2003 Server. MATLAB is used to solve the ILP formulations.

To evaluate our solution in terms of redundancy and number of p-cycle structures, we compare its performance to the heuristics CIDA and SLA introduced in [3], [8]. We also compare our algorithm to the exact solution given by an ILP formulation [1]. CIDA generates in the first step a limited set of candidate p-cycles, then it selects the set of p-cycles that protects the network using actual efficiency [3]. SLA generates also a set of candidate p-cycles, which is very small, and then the efficient sub-set of p-cycles protecting the network is selected using the ILP formulation. Note that in the optimal solution, all cycles are used as candidate p-cycles for the ILP.

TABLE II  
REDUNDANCY

Solutions Topologies	COST 239	KL
Optimal Solution	72.5%	78.2%
CIDA Solution	89.6%	91.8%
SLA Solution	98.4%	101%
Our basic Solution	83.9%	88.7%
Extension Solution	80.1%	86.3%

We can see in Table II that the value of redundancy obtained by our basic solution is better than the value obtained by the two heuristics CIDA and SLA for the two topologies. When we relax the constraint of the aggregation process by adding a new sub-condition, our solution becomes even more efficient. The redundancy of the improved solution is 7.6% and 8.1% far from the ILP solutions, respectively for the COST 239 and KL test networks. The reason for this efficiency (compared to well know solutions such as CIDA and SLA) is that in our approach, the construction of each p-cycle is done by incremental aggregation cycle and this last takes into account the working capacity, thus let us control the redundancy of each p-cycle that we construct. In other word we construct only cycles with small redundancy which lead to reduce the total redundancy of the network. When we relax the constraint of the aggregation we enlarge the search space in order to find

an efficient p-cycle.

TABLE III  
RUNNING TIME

Solutions	Topologies	COST 239	KL
Optimal Solution		127728(s)	11356(s)
CIDA Solution		0,95(s)	0.13(s)
SLA Solution		159(s)	8.2(s)
Our basic Solution		0.04(s)	0.08(s)
Extension Solution		0.08(s)	0.07(s)

We can see also that our running time is very little when we compare it with the others solution. The reason is that our algorithm does not go through a preliminary step of candidate p-cycles generation.

TABLE IV  
NUMBER OF P-CYCLES

Solutions	Topologies	COST 239	KL
Optimal Solution		23.4	16
CIDA Solution		27.5	18
SLA Solution		17	13
Our basic Solution		17,5	11
Extension Solution		15.4	10.5

Since the number of distinct p-cycle structures in the final set of p-cycles is an important performance for the network management task, we also evaluate our algorithm in terms of number of distinct p-cycles. Smaller number of structures in the final set of p-cycles means better performance. The simulation shows that our algorithm computes a small number of distinct structures. That is due to the criterion that we have introduced earlier. We can note that SLA generates a small number of structures in the COST239 topology when we compare it with our basic solution, because its set of candidate p-cycles is very small, but the value of redundancy obtained by SLA is significantly higher than the value obtained by our solution.

## V. CONCLUSION

In this paper, we addressed the protection in WDM optical networks using p-cycles. We studied the problem of finding an optimal set of p-cycles that protect the optical network for a given working capacity. We proposed an algorithm based on an incremental aggregation of cycles to construct the set of p-cycles which protect the network. Our solution constructs a set of p-cycles by taking into account the working capacity of the network and without going through a preliminary step of candidate p-cycles generation. This enables us to generate an efficient set of p-cycles without any consideration or assumption about the set of candidate p-cycles. To avoid the algorithm getting stuck in local minima we have proposed an extension to our solution by improving the conditions of the aggregation process. Simulation results showed that our solution has a good trade-off between resource utilization, running time and has a small number of p-cycles for the two test networks. Results also showed that the extended solution improve significantly the performances of our algorithm.

## REFERENCES

- [1] W. D. Grover and D. Stamatelakis, *Cycle Oriented Distributed Pre-configuration: Ring-like Speed with Mesh-like Capacity for Self-planning Network Restoration*, In Proc. of IEEE International Conference on Communications, pp. 537-543,1998.
- [2] D. A. Schupke, C. G. Gruber and A. Autenrieth, *Optimal Configuration of p-Cycles in WDM Networks*, In Proc. of IEEE International Conference on Communications, pp. 2761-2765, 2002.
- [3] 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*, In Proc. of the Fourth International Workshop on the Design of Reliable Communication Networks DRCN, pp. 212-220, 2003.
- [4] D. A. Schupke, *An ILP for Optimal p-Cycle Selection without Cycle Enumeration*, in Proc. of the Eighth Working Conference on Optical Network Design and Modelling, 2004.
- [5] Z. Zhang, W. Zhong and B. Mukherjee, *A Heuristic Algorithm for p-Cycles Configuration in WDM Optical Networks*, in Proc : Opto-Electronics and Communications Conference, pp. 568-569, 2003.
- [6] C. G. Gruber, *Resilient Networks with Non-Simple p-Cycles*, in Proc. of the International Conference on Telecommunications, 2003.
- [7] H. N. Nguyen, D. Habibi, V. Q. Phung, S. Lachowicz, K. Lo and B. Kang, *Joint Optimization in Capacity Design of Networks with p-Cycle Using the Fundamental Cycle Set*, In Proc. of IEEE GLOBECOM, 2006.
- [8] H. Zhang and O. Yang, *Finding Protection Cycles in DWDM Networks*, In Proc. IEEE International Conference on Communication, pp. 2756-2760, 2002.
- [9] C. Liu and L. Ruan, *Finding Good Candidate Cycles for Efficient p-Cycle Network Design*, In Proc. 13th International Conference on Computer Communication and Networks, pp. 321-326, 2004.
- [10] W. D. Grover and J. Doucette, *Advances in Optical Network Design with p-Cycles: Joint optimization and pre-selection of candidate P-cycles*, In Proc. of the IEEE-LEOS Summer Topical Meeting on All Optical Networking, pp. WA2-49-WA2-50, 2002.
- [11] D. P. Onguetou and W. D. Grover, *p-Cycle Network Design: from Fewest in Number to Smallest in Size*, In Proc of the 6th International Workshop on Design and Reliable Communication Networks, 2007.
- [12] J. L. Marzo, E. Calle, P. Vil, A. Urra, *Performance evaluation of minimum interference routing in network scenarios with protection requirements*, Computer Communications, pp. 3161-3168, 2007.
- [13] J. Van Leeuwen, *Algorithms and Complexity*, Handbook of Theoretical Computer Science, 1990.