MODELS AND HEURISTICS FOR P-CYCLE NETWORKS DESIGN

Agustín Pecorari
Depto. de Computación, FCEN- Universidad de Buenos Aires
Pabellón I- Ciudad Universitaria- Buenos Aires - Argentina
& Global Planning Tools at Verizon
agustin.pecorari@gmail.com

Irene Loiseau
Depto. de Computación, FCEN, Universidad de Buenos Aires
Pabellón I- Ciudad Universitaria- Buenos Aires - Argentina
irene@dc.uba.ar

ABSTRACT

A telecommunication network is said survivable if it is still able to provide communication between sites it connects after certain component fails. Mesh restoration schemes were widely used in the 1970s and 1980s. Ring based topologies were introduced in the late 80s. Around ten years later appeared the p-cycle networking concept. A p-cycle provides one protection path for a failed span it crosses and it also protects spans that have both end nodes on the cycle but are not themselves on the cycle. This technology is reported to simultaneously provide the switching speed and simplicity of rings with the efficiency and flexibility for reconfiguration of a mesh network. The basic problem we deal with may be seen as the problem of covering with p-cycles all the demands on a 2-connected graph minimizing the total cost. We present two MIP models that do not require a priori enumeration of candidate cycles.

Key words: telecommunications, survivable networks, p-cycles.

1. INTRODUCTION

Efficient protection is a major issue in telecommunications. Optical networks provide the backbone infrastructure for telecommunication networks. Because of the high-speed of optical networks, network survivability is of paramount importance. Upon an accidental failure such as a fiber-cut, it is imperative that the network can achieve fast optical recovery in order to minimize data loss.

A network is said survivable if it is operational even if certain component fails, that is, if it is still able to provide communication between sites it connects. Mesh restoration schemes were widely used in the 1970s and early 1980s. Ring based topologies were introduced in the late 80s based on self-healing rings (SHR) networks technology. Around ten years later appeared the p-cycle networking concept that enables fast span/link protection with high capacity efficiency. The idea is to organize the spare capacity in the network into a set of preconfigured cycles to protect working capacity at each span.

This new technology is reported to simultaneously provide the switching speed and simplicity of rings with the much greater efficiency and flexibility for reconfiguration of a mesh network. As this architecture is based on local protection actions Add Drop Multiplexers (ADM) are simple and cheap. This technology can be very efficient for several types of service. Asthana et al. (2010) in their review propose a classification of the different ways p-cycles networks provide protection.

A single unit capacity p-cycle is a cycle composed of one spare channel on each span it crosses. A span traversed by a p-cycle is called an **on-cycle** span of this p-cycle. If a span is not traversed by a p-cycle but its two end nodes are, then it is called a **straddle** span of this p-cycle. In Figure 1, **on-cycle** spans are marked in bold.

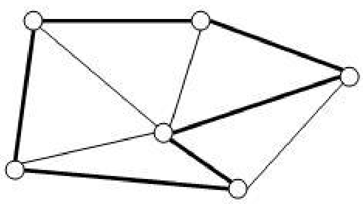

Figure 1. p-cycle

If an **on-cycle** span fails the p-cycle provides one protection path as shown on Figure 2. If a **straddle** span fails the p-cycle provides two protection paths as shown on figure 3.

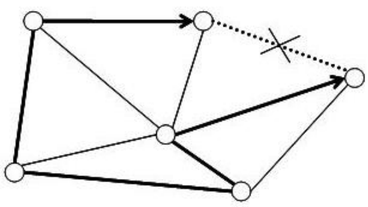

Figure 2. An example of on-cycle protection.

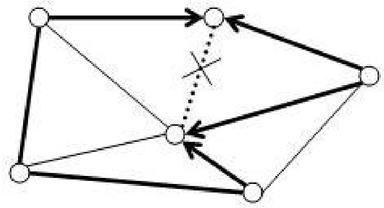

Figure 3. An example of straddle protection.

In this paper we deal with p-cycle network planning problems. The basic problem may be stated as the problem of covering with p-cycles all the demands on a 2-connected graph minimizing the total cost. This problem is called at the Spare Capacity Optimization (SCO) Problem (also appears at the literature as the Spare Capacity Placement Problem, SCP, or Spare Capacity Allocation Problem, SCA). Demands and capacity are expressed in number of channels. In SCO routing is a fixed input. We assume that traffic demands have already been routed, so they are known and fixed on each span. Schupke (2004) shows that the SCO problem is NP-hard by means of a simple reduction from the Hamiltonian cycle problem.

In the more general problem, the Joint Capacity Optimization (JCO), routing and spare capacity placement are jointly optimized to minimize the total capacity required. That is, demands are not assumed to be routed in advance. Solving JCP would produce better solutions but is much harder to solve that SCA.

Several heuristics, some of them based on MIP models, have been proposed for SCO problem. For reviews on p-cycle networks architectures, models and methods see Grover et al. (2005) and Kiaie et al. (2009).

1.1 ILP model for SCO

In what follows we assume that G=(V,X) is a 2 connected graph, n = |V| and m =|E|
Given a cycle, we can easily compute its cost and its straddles, and consequently we can determine the demands protected. So, if we were able to find the set of all possible cycles of the graph we could create every possible combination of cycles needed to cover the network demands. To find the optimal cover we would need to check which the cheapest combination is.

Theoretically we can find all the possible cycles of a graph using one of the classical algorithms, for example those of Johnson (1975) or Tarjan (1972) and we can formulate a simple ILP to choose the optimal cover. We present here a simplified version of one of the models mentioned in Grover et al (2005).

Coefficients:

***K*;** is the set of cycles of the graph.
p_{ik} encodes the protection relationship between the span *i* and eligible cycle *k*. So p_{ik} = 1 if the span *i* is on-cycle, p_{ik} = 2 if the span *i* is straddle, and p_{ik} = 0 otherwise.
d_i**;** demands on span *i*.
c_k**;** cost of each unit-capacity copy of cycle *k*.

Variables:

$x_k \geq 0$: Integer variable. This variable represents the number of unit-capacity copies of cycle *k* in the solution.

ILP Formulation:

$$\min \sum_{k \in K} c_k x_k \tag{1}$$

$$\sum_{k \in K} p_k^i x_k \geq d_i, \forall i \in E \tag{2}$$

As a graph may have an exponential number of cycles the model may have on the worst case an exponential number of variables. We know that the number of cycles in a complete graph of size n is:

$$\sum_{k=3}^{n} (k-1)! \frac{C_k^n}{2} \tag{3}$$

Even to handle the output of the algorithm that finds the cycles can be a daunting task.

Several heuristics have been developed based on this model, that only generate a subset of the cycles having promising indicators. Grover et al (2005), present a review of some of these methods. Here we mention two possible criteria:

- **Topological Score**: is the total number of spans the p-cycle is able to protect (demands are not taking into account in this case).

$$TS_k = \sum_{i \in E} p_k^i \tag{4}$$

- **A priori efficiency:** is the topological score divided by the p-cycle cost.

$$AE_k = \frac{TS_k}{c_k} \tag{5}$$

Liu and Ruan (2004) also propose heuristics to search for good candidate cycles. Their algorithm generates O(|E|) high efficiency cycles and O(|E|) short cycles. They claim that the cycles generated can lead to near optimal solutions when used by either ILP or a heuristic algorithm.

The rest of the work is organized as follows. In Section 2 we present two new MIP models that do not require candidate cycle enumeration. In Section 3 we present numerical results and Section 4 is devoted to conclusions.

2. ILP and MIP formulations for the SCO problem

Schupke (2004) presents an ILP model for the SCO without previous cycle enumeration. He assumes that a maximum number of cycles is known in advance. In an ILP formulation, cycles can be defined by requiring each node to have 2 or 0 on cycle spans incident to it. But with this definition multiple disjoint cycles may be obtained. Following Wu et al. (2010) notation we call the sets of cycles generated this way CS_j. So the CS_j may contain one or multiple disjoint sets. At Schupke model a predetermined number J of CS_j sets of disjoint cycles is generated and one cycle at each CS_j is determined by means of flow constraints. The number of variables of this model is $J(|V|^2 + |V||E| + 4|E| + |V| + 1)$ and $J(|V|^2 + |V||E| + 9|E| + |V| + 1) + |E|$ is the number of constraints, but it needs a long running time to obtain an optimal solution. So in the same paper a four-step heuristic is designed to obtain suboptimal solutions.

Wu et al. (2007, 2010), formulate three new models for the SCO based on Schupke previous work. They also consider a maximum of J set of cycles. At their first model they do not explicitly try to ensure a simple cycle at each set, instead they check if every span in the network can be protected by some CS_j. This model has $J|E|^2 + J|E||V| + J|V|$ variables and $J|E|(|E| + |V| + 2) + J|V| + |E|$ constraints. The model that provided the best results is called "cycle-exclusion" model. It is based on new constraints authors introduce, intended to ensure that only a single valid cycle from each CS_j will appear at the solution. They call these constraints electric voltage constraints (see section 2.2). It has $3J(|E| + |V|)$ variables and $4J|E| + 2J|V| + |E| + J$ constraints.

2.1. New IP models for SCO Problem

We formulated 4 new models for the SCO. Because of lack of space we are presenting here only two of them. At the two models not described here constraints derived from Miller-Tucker- Zemlim subtour elimination for the TSP were included in order to ensure only one cycle on each CS_j.. In the models presented here we first introduce a new set of constraints to establish which edges ar straddles on a p-cycle. Then we define new constraints based in the cycle space C(G) of a graph.

In both models the number of cycles to be generated is restricted to an arbitrary number J, as on the models on Schupke, (2004) and Wu et al, (2007, 2010). The minimum number of cycles needed to protect a network with p-cycles is $J \geq \left| D/2 \right|$ where D is the highest demand on any single edge of graph G. Also we can state an upper bound of the maximum number of cycles on an optimal solution, that is $J \leq \sum_{i \in E} d_i$, where d_i is the demand on edge i. Then we have:

$$\sum_{i \in E} d_i \geq J \geq \left\lceil \frac{D}{2} \right\rceil$$

This threshold can be tight in the worst case, as it can be seen on figure 4. In this graph there are cycles of length 3 with zero costs and demand equal one on one edge of each cycle and zero in the others. Each cycle is connected to the next one by an edge (bold edges) of infinity cost and zero demand:

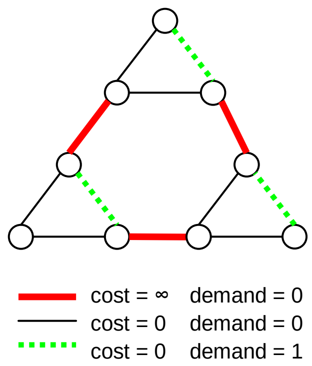

Figure 4. Upper bound on J.

The limitation of J cycles could be seen as a serious problem. But on real networks the optimum is not far from the lower bound (on the standard benchmark COST239 network D = 11, so $J \geq 6$ and the number of cycles on its 19 optimal solutions is 7). In real networks cost are very tightly related to distances and demands are not as sparse on networks were planners are really interested. And D is never too big because, in that case we should be planning on a bigger mux level.

2.1.1. MIP model for SCO problem based on voltage subtour constraints and straddle identification

As we mentioned above at their most compact model Wu et al (2010) define electric voltage constraints as a way to ensure a single cycle at each CS_j. Each span e is given a direction (or vector) e = (u,v) or e = (v,u) (but not both). If we have e=(u,v) we say that u is de head of the vector and v the tail. A root is a node which is the head of two vectors. A reversal node is a node that is the tail of two vectors. A value called voltage is assigned to each node in CS_j. We require the tail of a vector to have a larger value than its head. There is a single pair of root and node reversal nodes if and only if a feasible set of voltage values exists. This way ensuring a single cycle on each CSj is equivalent to ensuring only one pair of root an reversal nodes on CS_j, For more details about these constraints see Wu et al (2010).

We propose in this model a different representation of the straddles. If *M* is the incidence matrix of the graph *G* and if x represents a cycle, and then $r = \frac{1}{2}Mx$ represents the nodes of the cycle and can only take values in {0, 1}. Then, the result of $x(M^t r - 2I)$ is a vector whose elements can only take three values in {0, 1, 2}. The meaning of those values corresponds to the number of nodes (on the p-cycle) incident to each edge minus the on-cycle nodes. So, if the value of one element of $x(M^t r - 2I)$ is two, it is a straddle edge.

Sets, matrices and indices:

n: Total number of nodes of graph *G* (|*V*|).
m: Total number of edges of graph *G* (|*E*|).
J: The maximum number of cycles allowed in the p-cycle solution.
j: Cycle set index, j $\in$ {1, 2, …, *J*}.
***cost**$_e$*: Cost of adding one unit of spare capacity on edge *e*.
***d**$_e$*: Demand on edge *e*.
M: Incidence matrix of the graph *G* (*n* * *m*).
L: A very big number.
$\alpha = \frac{1}{|V|}$: Coefficient to guarantee feasible voltages.

Variables:

x^j_{eo}: Binary variable. It takes 1 if the edge e is on the generated cycle j and 0 otherwise. Index $o \in \{0,1\}$ indicates direction of the vector needed to define voltage constraints.
y^j_e: Binary variable. It takes 1 if the edge e is a straddle of the generated cycle *j* and 0 otherwise.
$root^j_q$: Binary variable. It takes value 1 if node u is a root node on CS_j and 0 otherwise.
$volt^j_q$: Continuous variable. Is the voltages value of node u.

Formulation:

$$\min \sum_{j\in J}\sum_{e\in E} \cos t_e (x^j_{e,0} + x^j_{e,1}) \tag{6}$$

s.t.

$$\sum_{e=uv\in E}(x^j_{e0} - x^j_{e1}) = 0, \forall u \in V, \forall j \in J \tag{7}$$

$$\frac{1}{2}(x^j_0 + x^j_1)(\frac{1}{2}M^t M - 2I) \geq y^j, \forall j \in J \tag{8}$$

$$x^j_{e0} + x^j_{e1} \leq 1, \forall e \in E, \forall j \in J \tag{9}$$

$$M(x^j_0 + x^j_1) \leq 2, \forall j \in J \tag{10}$$

$$\sum_{u\in V} root^j_u = 1, \forall j \in J \tag{11}$$

$$volt^j_v - volt^j_u \geq \alpha x^j_{eo} - (1 - x^j_{eo}) - root^j_v L, \forall e = (u,v) \in E, \forall j \in J \tag{12}$$

$$d_e \leq \sum_{j\in J}(x^j_{e0} + x^j_{e1} + 2y^j_e), \forall e \in E \tag{13}$$

$$x^j_{eo} \in \{0,1\}, \forall j \in J, \forall e \in E, o \in \{0,1\} \tag{14}$$

$$y^j_e \in \{0,1\}, \forall j \in J, \forall e \in E \tag{15}$$

$$root^j_u \in \{0,1\}, \forall j \in J, \forall u \in V \tag{16}$$

$$volt^j_u \in [0,\infty), \forall j \in J, \forall u \in V \tag{17}$$

The total cost of all p-cycles in minimized by (6).

Constraints (7) state flow conservation. Constraints (8) are used to indentify spans that are straddle, as it was explained above. Constraints (9) state that an edge can take only one direction.

Constraints (10) restrict nodes to have only degree less or equal to 2. Constraints (11) force a root to exist on each cycle. Constraints (12) state difference of voltage on each span with the exception of the last span on the cycle. Constraints (13) impose that the demands have to be satisfied.

Number of variables: J(3|E| + 2|V|).
Number of constraints: 4J|E| + 2J|V| + |E| + J.

So our model has less variables and exactly the same number of constraints that the cycle exclusion model of Wu et al (2010).

2.1.2 MIP model for SCO problem based on cycle space and voltage subtour constraints

Cycles of a graph can be represented by binary arrays, where the i^{th} position indicates if the i^{th} edge of the graph is part of the cycle or not.

The cycles and the union of edge disjoint cycles of a graph forms a space that we will call *C = C(G)* (see for example Diestel, 2000). The cycle space *C(G)* is the subspace of *E(G)* spanned by all the cycles in *G*, more precisely, by their edge sets. The dimension of *C(G)* is the cyclomatic number $\mu = |E| - |V| + c$, where c is the number of connected components of the graph. On 2-connected graphs (as is our case) c is 1.

If we consider an arbitrary spanning tree T of *G*, then each edge not in T determines a unique cycle if it is added to the spanning tree. These cycles are called fundamental cycles of G. The set of all fundamental cycles with respect to an arbitrary spanning tree is a base of the cycle space *C(G)*.

Following Bapeswara Rao and Murty (1969) we call the disjoint union (ring union, mod 2 addition, or "exclusive or") operation between cycles vectors a circ. So a circ is either a cycle, the null cycle or an edge-disjoint union of cycles.

We propose a new model for the SCO using these concepts. We will need to calculate a priori a spanning tree of the graph and to determine the corresponding set of fundamental cycles.

We replace constraints (10) of model on section 2.1.1 by constraints (22). They state that each generated cycle is obtained by means of mod 2 addition of fundamental cycles. Voltage constraints are included to avoid edge-disjoint union of cycles. Using the same notation as in 2.1.1, we have:

Matrices:

B: Matrix of a set of fundamental cycles ($m * \mu$). Each fundamental cycle is a column and each row is an edge.

Variables:

x^j_{eo}: Binary variable. It takes 1 if the edge e is on cycle *j* and 0 otherwise. Index $o \in \{0,1\}$ indicates direction of the vector needed to define voltage constraints.
c^j_i: Binary variable. It takes 1 if cycle j is generated by fundamental cycle *i* and 0 otherwise.
y^j_e: Binary variable. It takes 1 if the edge e is a straddle of the generated cycle *j* and 0 otherwise.
p^j_e: Integer variable. Is an auxiliary variable that indicates how many pairs of fundamental cycles are selected to generate the cycle *j* that includes the edge *e*.
$root^j_u$: Binary variable.
$volt^j_u$: Continuous variable.

Formulation:

$$\min \sum_{j\in J}\sum_{e\in E} \cos t_e (x^j_{e0} + x^j_{e1}) \qquad (18)$$

s.t.

$$\sum_{e=uv\in E} (x^j_{e0} - x^j_{e1}) = 0, \forall u \in V, \forall j \in J \qquad (19)$$

$$\frac{1}{2}(x^j_0 + x^j_1)(\frac{1}{2}M^tM - 2I) \geq y^j, \forall j \in J \qquad (20)$$

$$x^j_{e0} + x^j_{e1} \leq 1, \forall e \in E, \forall j \in J \qquad (21)$$

$$Bc^j = 2p^j + (x^j_{e0} + x^j_{e1}), \forall j \in J \qquad (22)$$

$$\sum_{u\in V} root^j_u = 1, \forall j \in J \qquad (23)$$

$$volt^j_v - volt^j_u \geq \alpha x^j_{eo} - (1 - x^j_{eo}) - root^j_v L, \forall e = (u,v) \in E, \forall j \in J \qquad (24)$$

$$d_e \leq \sum_{j\in J}(x_{e0}^j + x_{e1}^j + 2y_e^j), \forall e \in E \quad (25)$$

$$c_i^j \in \{0,1\}, \forall j \in J, \forall i \in C \quad (26)$$

$$p_e^j \in \{0,\infty\}, \forall j \in J, \forall e \in E \quad (27)$$

$$x_{eo}^j \in \{0,1\}, \forall j \in J, \forall e \in E, o \in \{0,1\} \quad (28)$$

$$y_e^j \in \{0,1\}, \forall j \in J, \forall e \in E \quad (29)$$

$$root_u^j \in \{0,1\}, \forall j \in J, \forall u \in V \quad (30)$$

$$volt_u^j \in [0,\infty), \forall j \in J, \forall u \in V \quad (31)$$

The objective function (18) and constraints (19) to (21) and (23) to (25) have been defined in the previous model (constraints (7) to (9) and (11) to (13)). Constraints (22) generates cycles using the base of the cycle space as it was explained above.

Number of variables: 5J|E| + 2J|V|.
Number of constraints: 5J|E| + 2J|V| + J + |E|.

3. Numerical results

We implemented our models and the best model from Wu et al (2010) in order to compare their efficiency. In all the cases we used as value for J. They models implemented on C++ using Concert Technologies from ILOG-IBM CPLEX 10.2. The server runs under HP-Unix on 16 processors at 1.1 GHz and 8 Gb RAM. The p-cycle community uses the COST 239 (figure 5) network that represents European telecommunications connections as standard benchmark case.

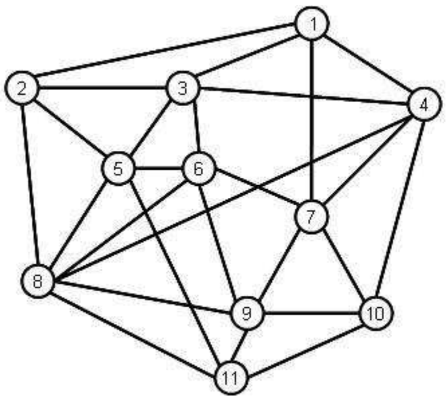

Figure 5. "Cost239"

We know the objective value of an optimal solution of SCO for COST 239 is 32,340. Table 1 shows the numerical results of the three implemented models with a run time limit of 400 seconds. Reduced MIP refers to the number of rows and columns of the problem after the pre-solve procedure of CPLEX has been applied. WYH stands for the cycle exclusion model of Wu et al(2010), DV is the model in section 2.1.1 and CBV the model presented in section 2.1.2.

Model	Best solution	Optimality gap (%)	Reduced MIP Rows	Reduced MIP Columns	Reduced MIP non-zero coefficients
WYH	32,390	1.02	915	777	3871
DV	32,930	4.70	913	686	5229
CBV	32,340	0.00	913	686	5229

Table 1. "Cost239" results after 400s of execution time.

Note that even if the initial number of variables and constraints of model presented in section 2.1.2 are not the best among these models, the final numbers of variables and constraints (Reduced MIP) are the same as those of the most compact model.

We also tested all the models on a set of networks representing real USA telecommunications networks (data available from authors). The set consists of 15 networks with number of nodes ranging from 6 to 13 and edges ranging from 15 to 78. The stopping criterion was to obtain an optimality gap of 5% or less. Results are summarized on tables 2 and 3.

		Objective function				Gap		
Network	**# Cycles**	**WYK**	**DV**	**CBV**	**Optimum**	**WYK**	**DV**	**CBV**
VZ_US_PIP_001	106967	33.345	33.320	32.550	32.240	3,43%	3,35%	0,96%
VZ_US_PIP_002	37286	37.370	38.070	37.370	37.370	0,00%	1,87%	0,00%
VZ_US_PIP_003	37286	35.370	35.770	35.370	35.170	0,57%	1,71%	0,57%
VZ_US_PIP_004	37286	41.060	41.095	40.670	39.980	2,70%	2,79%	1,73%
VZ_US_PIP_005	15341	25.407	26.059	25.472	24.937	1,88%	4,50%	2,15%
VZ_US_PIP_006	15341	26.190	26.425	26.190	26.190	0,00%	0,90%	0,00%
VZ_US_PIP_007	15341	30.534	30.549	30.534	30.534	0,00%	0,05%	0,00%
VZ_US_PIP_008	15341	33.277	33.532	33.216	32.736	1,65%	2,43%	1,47%
VZ_US_PIP_009	15341	37.306	37.521	37.256	37.071	0,63%	1,21%	0,50%
VZ_US_PIP_010	15341	45.421	46.251	45.871	44.084	3,03%	4,92%	4,05%
VZ_US_PIP_011	3354	43.936	43.881	43.591	42.054	4,48%	4,34%	3,65%
VZ_US_PIP_012	1636	37.201	37.401	35.754	35.754	4,05%	4,61%	0,00%
VZ_US_PIP_013	167	30.184	30.654	29.984	29.984	0,67%	2,23%	0,00%
VZ_US_PIP_014	70	14.740	14.740	14.740	14.740	0,00%	0,00%	0,00%
VZ_US_PIP_015	18	14.775	14.775	14.775	14.775	0,00%	0,00%	0,00%

Table 2. Results on a set of USA networks

Model	Avg. time in seconds
WYH	356
DV	896
CBV	232

Table 3. Average running times

4. Conclusions

Our primary focus was to propose new MIP models that do not require candidate cycle enumeration for p-cycles networks design problems. Two new MIP model were formulated for solving the spare capacity optimization (SCO) problem. All the models have a polynomial number of variables and constraints and if the value of J is big enough, the exact optimal solution can be obtained.

Our main contributions are that cycle generation is done by means of a constraint based on cycle space properties and that the straddle representation is based on the incidence matrix of the network. These concepts lead to very compact models and efficient algorithms.

References

Asthana R., Singh Y.N. and Grover W.D. (2010). P-cycles: an overview. IEEE Communications Surveys & Tutorials, 12 (1), 97-111.

Bapeswara Rao V. V. and Murti V. G. K. (1969) Enumeration of all circuits of a graph. Electronics Letters, 5, 282-283.

Diestel R. , Graph Theory. Springer-Verlag New York, 2000.

Grover W. D., and Kodian A. (2005) Failure-independent path-protection with p-cycles: efficient, fast and simple protection for transport optical networks. Journal of Lightwave Technology, 23 (10), 3241- 3258.

Grover W, Doucette J., Kodian A., Leung,D. ,Sack,A., Clouqueur,N. and Shen,G. Design of networks based on p-cycles, *in* Handbook of Optimization on Telecommunications, Resende,M. and Pardalos, P.(eds). Springer, 2005.

Johnson D. B. (1975) Finding all the elementary circuits of a directed graph. SIAM Journal on Computing. 4 (1), 77-84.

Kiaei, M.S., Assi,C., Jaumard, B., (2009) A survey on the p-cycle protection method, IEEE Communications Survey Tutorials, 11 (3), 53-70.

Liu C. and Ruan L. (2004) Finding good candidate cycles for efficient p-cycle network design. Proceedings. 13th International Conference on Computer Communications and Networks. ICCCN 2004.

Sack A. and Grover W.D. (2004) Hamiltonian p-cycles for fiber-level protection in homogeneous and semi-homogeneous optical networks, IEEE Networks, 18 (2), 49-56.

Schupke D. A. (2004) An ILP for optimal p-cycle selection without cycle enumeration. Proceedings of the IFIP 8th Conference on Optical network Design and Modelling (ONDM '04).

Tarjan R. (1972) Enumeration of the elementary circuits of a directed graph. Technical Report, Department of Computer Science. Cornell University.

Wu B., Yeung K. L., and Ho P.H. (2010) ILP formulations for p-cycle design without candidate cycle enumeration, IEEE/ACM Transactions on Networking, 18 (1), 284-295.

Wu B., Yeung K. L., Liu, K.S. and Xu, S. (2007) A new ILP-based p-cycle construction algorithm without candidate cycle enumeration. Proceedings IEEE ICC 2007, 2236-2241.

Wu B., Yeung K. L. and Xu, S. (2007) ILP formulation for p-cycle construction based on flow conservation. Proceedings IEEE GLOBECOM 2007, 2310-2314.