MODELS FOR COMMUNICATION NETWORK DESIGN WITH SURVIVABILITY REQUIREMENTS

Remberto Emanuel Delgadillo
Depto. de Computación, FCEN - Universidad de Buenos Aires
Pabellón I, Ciudad Universitaria, Buenos Aires, Argentina
emanuel.dc0@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 service after one of its components fails. In the late 90's the concept of p-cycles appeared as a promising alternative to previous technologies (mesh and ring). 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. P-cycle based networks gather the best characteristics of both mesh and ring based topologies: restoration speed and efficient use of spare capacity. The objective of the Spare Capacity Allocation problem (SCA) is to protect all the traffic of a network with p-cycles, minimizing the total cost. Here we propose a new ILP model for SCA that does not require a priori candidate cycle enumeration. Results improved those of previous models.

KEYWORDS. survivable networks, p-cycles, integer linear programming models.

1. Introduction

Telecommunications industry has always been a source of optimization problems (Resende and Pardalos (2006)), including those related to network survivability. A network is said survivable if it is able to provide communication between sites it connects, even if certain component fails. Networks based on 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. The main advantage of mesh-based survivable networks is the efficient use of spare capacity, but they require long time to achieve restoration. On the other hand, ring-based networks achieve restoration faster because of the simple mechanism they use to do that. But their disadvantage is that they need to have much more spare capacity than mesh networks.

The p-cycle networking concept was introduced in 1998 by Grover and Stamatelakis (1998). The idea is to organize the spare capacity in the network into a set of preconfigured cycles to protect working capacity at each span. P-cycles based topologies enable fast span/link protection with high capacity efficiency. They simultaneously provide the switching speed and simplicity of rings and the much greater efficiency and flexibility for reconfiguration of a mesh network.

A single unit capacity p-cycle is a simple 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.

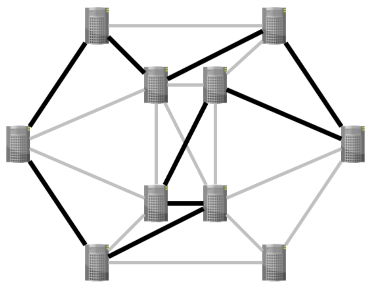

Figure 1: A p-cycle

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.

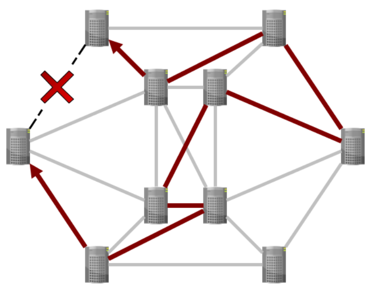

Figure 2: Example of on-cycle protection

In this work we deal with the Spare Capacity Allocation (SCA) problem. The network spare capacity has to be designed by covering with p-cycles all the demands on a 2-connected graph minimizing the total cost (the same simple cycle can be used more than once if necessary). This problem also appears at the literature as the Spare Capacity Placement Problem, SCP, or Spare Capacity Optimization Problem, SCO. Demands and capacity are expressed in number of channels.

We assume that traffic demands have already been routed, so they are known and fixed on each span. The cost of adding one unit of capacity to each span is provided. Schupke (2004) shows that the SCA problem is NP-hard by means of a simple reduction from the Hamiltonian cycle problem.

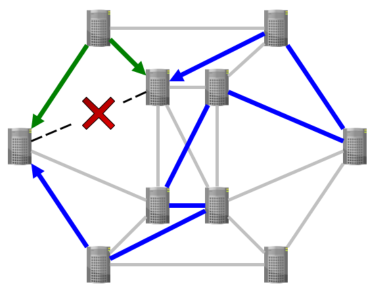

Figure 3: Example of straddle protection

In Section 2 related work is reviewed. In Section 3 a new mixed integer linear programming model for SCA is introduced. Inequalities intended to avoid symmetry and other valid inequalities are presented in Section 4. Computational results are given in Section 5 and Section 6 is devoted to concluding remarks.

2. Related work

A review of SCA and other p-cycle related problems can be found in Grover *et al*. (2006). We present here a simple model for SCA.

Notation

- $G = (V, E)$ is a 2-connected graph representing the network.
- P is the set of simple cycles of G.
- p^i_j encodes the protection relationship between the span i and eligible cycle j. So $p^i_j = 1$ if the span i is on-cycle, $p^i_j = 2$ if the span i is straddle, and $p^i_j = 0$ otherwise.
- d_i is the demand on span i.
- C_j is the cost of each unit-capacity copy of cycle j.

Variables

$x_j (\geq 0)$: Integer variable. This variable represents the number of unit-capacity copies of cycle j in the solution.

ILP Formulation

$$\min \sum_{j \in P} C_j x_j \qquad (1)$$

$$\text{s.t.} \sum_{j \in P} p^i_j x_j \geq d_i, \forall i \in E \qquad (2)$$

The objective function represents the sum of costs of all cycles in the solution. Constraints ensure that every working demand is protected.

As a graph may have an exponential number of cycles, this model may have on the worst case an exponential number of variables.

Several heuristics based on this model were proposed. Only a subset of all simple cycles (candidate or eligible cycles) is selected and the ILP problem is solved using this subset of variables.

Some criteria to evaluate candidate cycles sets are defined in Grover and Doucette (2002). Two of them are the topological score (TS) and a-priori efficiency (AE):

$$TS_j = \sum_{i \in E} p^i_j \qquad \forall j \in P \tag{3}$$

$$AE_j = \frac{TS_j}{C_j} \qquad \forall j \in P \tag{4}$$

When using these criteria a set of candidate cycles A is considered better than another set B if it has a higher average AE. Doucette *et al.* (2003) and Zhang and Yang (2002) propose candidate selection heuristics that select a polynomial number of candidate cycles, which are evaluated with the above mentioned score.

Schupke (2004) presents an ILP model for SCA without previous candidate cycle enumeration. He assumes that a bound J of the number of cycles in an optimal solution 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 each CS_j may contain one or multiple disjoint cycles. 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.* (2010) formulate three new models for SCA based on Schupke previous work. 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. These constraints are very similar to Miller-Tucker-Zemlin (MTZ) subtour elimination constraints for the Traveling Salesman Problem. The model has $3J(|E|+|V|)$ variables and $4J|E|+2J|V|+|E|+J$ constraints.

In Pecorari and Loiseau (2012) two new models are presented also assuming that a bound J for the number of cycles in an optimal solution is known. The model with best prefomance determines cycles using cycle basis concepts and MTZ subtour elimination. A basis of the Cycle Space of a graph is obtained from a spanning tree and every cycle of the graph can be obtained as a disjunctive union of cycles of the basis. This model has $5J|E| + 2J|V|$ variables and $5|E| + 2J|V| + J + |E|$ constraints.

3. A new formulation for SCA Problem

In what follows we will present a new ILP model for SCA that does not require previous candidate cycle enumeration. As in the models mentioned above we assume that we will have at most J cycles at the optimal solution. We include constraints defining cycles by requiring each node to have 2 or 0 on cycle spans incident to it, and we add subtour elimination constraints that are flow constraints similar to those introduced by Gavish and Graves (1978) for the Traveling Salesman Problem.

We add two nodes to the original graph G, a source s and a sink t. We also add arcs from s to every node in V and from nodes in V to t. Constraints in our formulation are determined by the following rules for each cycle of the solution:

- Flow is allowed from s to at most one node in G.

- Flow is allowed between adjacent nodes u and v in G in any direction, but only if the the edge (u, v) is selected to be part of the cycle.
- A binary flow is allowed from any node in G towards t.
- An edge (u, v) is protected (on-cycle or straddle) by the cycle if and only if both u and v have a flow of 1 towards t.

An example of the previous rules can be seen at Figure 4. The edges in bold belong to the cycle. Flow goes from every node in that cycle to t.

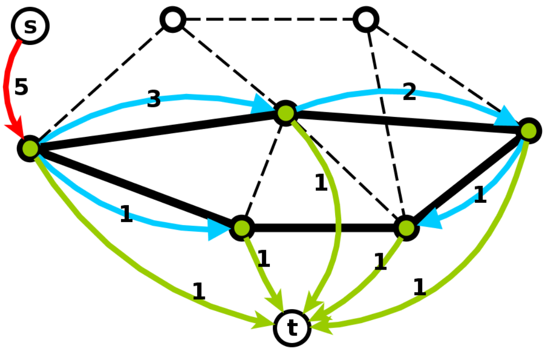

Figure 4: A cycle of the solution in our formulation

Notation and input data

$c_{(u,v)}$: cost of adding a unit of capacity on edge (u, v)

$d_{(u,v)}$: working demand on edge (u, v)

J : maximum number of cycles in the solution.

j : index for cycles ($1 \leq j \leq J$).

$N_G(u)$: set of nodes v such that $(u, v) \in E$

It is important to note that G is an undirected graph, so an edge incident to nodes u and v can be noted as (u, v) or (v, u) indistinctly.

Variables

$z^j_{(u,v)}$: Binary. $\forall (u, v) \in E$. Equals to 1 if and only if edge (u, v) belongs to cycle j.

$y^j_{(u,v)}$: Real (≥ 0). $\forall (u, v) \in E$. Represents the amount of capacity units of protection that cycle j gives to edge (u, v). This variable has to take integer values (0, 1 or 2), but we can let it to be a real number since it will be always forced by the rest of the model to take an integer value.

x^j_{uv} : Real (≥ 0). $\forall u \in V, v \in N_G(u)$ and for $u = s, v \in V$. Represents flow in the direction $u \to v$ for cycle j. These variables and the following ones are necessary to guarantee that variables $z^j_{(u,v)}$ form at most one cycle for each index j.

b^j_u : Binary. $\forall u \in V$. Represents flow in the direction $u \to t$ for cycle j. Alternatively, this variable can be defined as being equal 1 if and only if node u belongs to cycle j.

r^j_u : Binary. $\forall u \in V$. Equals 1 if and only if flow from s to u is allowed.

The MIP Model

$$min \sum_{j} \sum_{(u,v)\in E} c_{(u,v)} z^j_{(u,v)} \tag{5}$$

s.t.

$$\sum_{v\in N_G(u)} z^j_{(u,v)} = 2 \times b^j_u \qquad \forall j \leq J, \forall u \in V \tag{6}$$

$$\sum_{v\in N_G(u)\cup\{s\}} x^j_{vu} - \sum_{v\in N_G(u)} x^j_{uv} = b^j_u \qquad \forall j \leq J, \forall u \in V \tag{7}$$

$$x^j_{uv} + x^j_{vu} \leq (|V| - 1) \times z^j_{(u,v)} \qquad \forall j \leq J, \forall (u,v) \in E \tag{8}$$

$$\sum_{u\in V} r^j_u \leq 1 \qquad \forall j \leq J \tag{9}$$

$$x^j_{su} \leq |V| \times r^j_u \qquad \forall j \leq J, \forall u \in V \tag{10}$$

$$y^j_{(u,v)} \leq 2 \times b^j_w \qquad \forall j \leq J, \forall (u,v) \in E, \forall w \in \{u,v\} \tag{11}$$

$$y^j_{(u,v)} \leq 2 - z^j_{(u,v)} \qquad \forall j \leq J, \forall (u,v) \in E \tag{12}$$

$$\sum_{j\leq J} y^j_{(u,v)} \geq d_{(u,v)} \qquad \forall (u,v) \in E \tag{13}$$

$$z^j_{(u,v)} \in \{0,1\}, \;\; y^j_{(u,v)} \in \mathbb{R}_+ \qquad \forall j \leq J, \forall (u,v) \in E$$

$$x^j_{uv} \in \mathbb{R}_+ \qquad \forall j \leq J, (\forall u \in V, v \in N_G(u)) \wedge (u = s, v \in V)$$

$$b^j_u \in \{0,1\}, \;\; r^j_u \in \{0,1\} \qquad \forall j \leq J, \forall u \in V$$

The objective function (5) is intended to minimize the cost of the solution. With (6) cycles are defined. But these constraints only states that each node has degree 0 or 2 at the solution, so it is necessary to add subtour elimination constraints. Constraints (7) ensure flow conservation at each node. Constraints (8) guarantee that flow is allowed only at the edges that belong to the cycle. $(|V| - 1)$ could be replaced by any bigger value. $(|V| - 1)$ is the minimum value that allows cycles of length less or equal to $|V|$ (necessary to allow all simple cycles). Constraints (9) ensure that, for each cycle, only one node is allowed to receive flow from s, and (10) allow the existence of flow from s towards that node. Coefficient $|V|$ can be replaced for a bigger value. $|V|$ allows every simple cycle to be part of the solution. Constraints (11) and (12) impose limits for the protection given by a cycle:

- If one of the nodes of an edge is not part of the cycle, protection in that edge must be 0. Otherwise, it will be at most 2 (constraints (11)). Note that there are two constraints (11) for each cycle and each edge.

- If an edge does not belong to the cycle, its protection will be at most 2 (*straddle*). Otherwise, it will be equal to 1. This is determined by constraints (12).

Constraints (13) ensure that enough protection is provided by all cycles at the solution to cover all demands.

The number of variables and constraints at our model is similar to the ones in Wu *et al.* (2010) and Pecorari and Loiseau (2012). It grows linearly respect to J and the size of the graph. But we have a lower number of integer variables. This is shown in Table 1. Our model is referred as MF. The others, WYH y CB, are the best reported models of Wu *et al.* (2010) and Pecorari and Loiseau (2012) respectively.

Model	Variables	Integer variables	Constraints
WYH	$3J(\|E\| + \|V\|)$	$3J\|E\| + 2J\|V\|$	$4J\|E\| + 2J\|V\| + J + \|E\|$
CB	$5J\|E\| + 2J\|V\|$	$5J\|E\| + J\|V\|$	$5J\|E\| + 2J\|V\| + J + \|E\|$
MF	$3J\|E\| + 2J\|V\|$	$J\|E\| + 2J\|V\|$	$4J\|E\| + 3J\|V\| + J + \|E\|$

Table 1: Comparison with other formulations.

4. More inequalities

In this section we will present some additional constraints we introduced to improve the running time of the model in previous section.

1. We first note that our model allows multiple symmetric solutions. Taking into account the possible values that integer variables may take, we found two classes of symmetry:

   (a) **Source symmetry:** For each cycle at the solution, only one variable r_u^j corresponding to the cycle nodes can be equal to 1. However, exchanging values between that variable and other variable r_u^j corresponding to other cycle node will result in an equivalent solution. This can be avoided forcing a priority on variables r_u^j by adding the following constraints to the model:

   $$u \times r_u^j \leq u - \sum_{v \in V, v < u} b_v^j \qquad \forall j \leq J, \forall u \in V \tag{14}$$

   Note that we assume that the nodes of the graph have labels ranging from 1 to $|V|$.

   (b) **Permutation symmetry:** Variables with same index j correspond to the same cycle at a solution. So, given a solution, another equivalent solution can be obtained exchanging all values between variables corresponding to two different cycles. This means that in the worst case, starting from a solution with J different cycles other $(J! - 1)$ symmetric solutions can be found. We added constraints to avoid this kind of symmetry, but early experiments showed that this leads to bad performance.

   It is very important to note that symmetry is not an exclusive issue of our model. Both Wu *et al.* (2010) and Pecorari and Loiseau (2012) best reported models have these two kinds of symmetry and another one that we avoid with our formulation. Also all other models without cycle enumeration we mentioned have at least permutation symmetry.

2. The following valid inequalities can be derived from the formulation:

   $$\sum_{u \in V} x_{su}^j = \sum_{u \in V} b_u^j \qquad \forall j \leq J \tag{15}$$

   Adding them to the model showed that preprocessing get stronger results, bounds are improved and overall performance is also improved.

3. We also defined the following valid inequalities that were used as user cuts within the frame of the ILP solver:

$$y^j_{(u,v)} \leq \frac{1}{2} \times (\sum_{(u,w)\in E, w\neq v} z^j_{(u,w)} + \sum_{(v,w)\in E, w\neq u} z^j_{(v,w)}) \qquad \forall j \leq J, \forall (u,v) \in E \qquad (16)$$

These inequalities state that edge (u, v) gets protection from cycle j only if there are other edges incident to u and v that form part of the solution. Including them leads to better linear relaxations.

5. Computational results

Computational experiments were carried on a PC Intel Core 2 Duo 1.8 GHz with 4GB RAM. Solver was ILOG-IBM CPLEX 12.4. Our test instances are the standard benchmark COST_239 and other USA real networks ranging from 6 to 13 nodes and from 15 to 39 edges (available from authors).

At Pecorari and Loiseau (2012) authors report that their results were better than those of Wu *et al*. (2010) best model when both were implemented using CPLEX 10.2 on the benchmark mentioned above. But when implementing the same models with CPLEX 12.4 we found out that Wu *et al*. (2010) model outperformed the model from Pecorari and Loiseau (2012). For example, for the COST_239 instance they reported a gap of 0% for their model and 1.02% for Wu *et al*. (2010) model with a runtime limit of 400 seconds (on other computing platform), but we got for the same instance 3.5% and 0.85% respectively after 100 seconds of execution. So we will compare efficiency with Wu *et al*. (2010) model.

In order to test the efficiency of the models, all the cycles of the networks were determined using the algorithm proposed in Grover (2003) and the optimal solution of each SCA problem was obtained by solving the ILP described on Section 2. We use the number of p-cycles on this optimal solution as value of J in our model. It is interesting to note that in these real problems although the number of cycles of the graphs varies between 18 and 106967, the values of J at optimal solutions ranges between 6 and 8.

As we mentioned before, we added to our model constraints (14) and (15). We also set inequalities (16) as user cuts of CPLEX 12.4. We had previously tested them as part of the formulation, but we got poorer results.

A key configuration to improve running time is setting priorities to integer variables for branching. A possible proper order would set highest priority to variables r^j_u. When a variable r^j_u is set to 1, all other variables r^j_v with same index j and b^j_w such that $w < u$ are forced to be 0 (because of source symmetry constraints). Secondly, variables b^j_u should have higher priority than the $z^j_{(u,v)}$ because they determine the nodes that will be used and force many $z^j_{(u,v)}$ variables to be 0. We did other experiments setting different priorities and the one that gave the best results is the following:

- Same priority to all r^j_u variables.
- Same priority to all b^j_u variables, but lower than the previous one.
- Same priority to all $z^j_{(u,v)}$ variables, but lower than the previous one.

It is important to mention that we also made some additional experiments with the model of Wu *et al*. (2010). We added to their formulation constraints to avoid symmetry and we tested several priority values for their integer variables but that did not improve running times.

	Objective Function			Gap (%)		Time (sec.)	
Instance	Optimum	MF	WYH	MF	WYH	MF	WYH
COST_239	32340	32340	32390	0	0.85	79.6	100
VZ_US_PIP_001	32240	32240	32460	1.24	4.59	100	100
VZ_US_PIP_002	37370	37370	37370	0	0.48	2.2	100
VZ_US_PIP_003	35170	35170	35170	0	0.43	33.7	100
VZ_US_PIP_004	39980	39980	40920	0.77	8.25	100	100
VZ_US_PIP_005	24937	24937	24937	0	5.06	27.5	100
VZ_US_PIP_006	26190	26190	26190	0	0.02	17.59	100
VZ_US_PIP_007	30534	30534	30534	0	0.02	0.93	100
VZ_US_PIP_008	32721	32721	33557	0	4.83	82	100
VZ_US_PIP_009	37071	37071	37071	0	0.42	47.2	100
VZ_US_PIP_010	44084	44084	44084	0	0	6.85	98
VZ_US_PIP_011	42054	42054	42054	0	0.02	34.3	100
VZ_US_PIP_012	35754	35754	35754	0	0.5	7.79	100
VZ_US_PIP_013	29984	29984	29984	0	1.17	4.45	100
VZ_US_PIP_014	14740	14740	14740	0	0	0.49	10.13
VZ_US_PIP_015	14775	14775	14775	0	0	0.17	4.16

Table 2: Comparison between our model and the model from Wu *et al.* (2010)

Main computational results are shown in Table 2. Our model is named MF and the other is referred as WYH. In both cases we set a limit of 100 sec. for all the runs. It is clear that we got much better results with our model. More detailed experiments are reported in Delgadillo (2013).

We also made additional experimentation setting different values for CPLEX parameters, including:

- Node selection strategies.
- Variable selection strategies.
- Cuts application frequency (CPLEX default cuts).
- Emphasis.
- Probing.
- CPLEX heuristics frequency.
- Conventional B&C instead of Dynamic Search.

Changing the default configuration of these parameters did not improve the performance of our model.

6. Conclusions

Our primary focus was to solve the SCA problem in a way that a priori candidate cycle enumeration was not required. To accomplish that, we proposed a new MIP model. This model has a polynomial number of variables and constraints and if the value of J is big enough, the exact optimal solution can be obtained. In practice, in real world networks, the value of J is pretty small, so it is possible to use this model as an heuristic with very good results.

We also made efforts to improve the performance of our model. We avoided part of the multiple symmetric solutions and introduce some inequalities that were used as cuts. In particular, symmetry is a common issue in all similar models we mentioned above.

Computational results show that we achieved much better running times than previous MIP models for SCA without cycle enumeration.

References

Delgadillo, R. E. Modelos y algoritmos para diseño de redes de comunicaciones con requisitos de supervivencia. Tesis de Licenciatura, Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2013.

Doucette, J., He, D., Grover, W. D. and Yang, O. (2003), Algorithmic Approaches for Efficient Enumeration of Candidate p-Cycles and Capacitated p-Cycle Network Design. *Fourth International Workshop on Design of Reliable Communication Networks, 2003. (DRCN 2003). Proceedings.*

Gavish, B. and Graves, S. C. (1978), The Travelling Salesman Problem and Related Problems. *Operations Research Center Working Paper;OR 078-78.*

Grover, W. D. *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice Hall PTR, 2003.

Grover, W. D. and Doucette, J. (2002), Advances in Optical Network Design with p-Cycles: Joint Optimization and Pre-Selection of Candidate p-Cycles. *Proceedings of IEEE/LEOS Summer Topicals 2002, Mont Tremblant, PQ, pp.49-50 (paper WA2).*

Grover, W. D., Doucette, J., Kodian, A., Leung, D., Sack, A., Clouqueur, M. and Shen, G. Design of survivable networks based on p-cycles. **Resende, M. and Pardalos, P.** (Eds.), *Handbook of Optimization in Telecommunications*. Spinger Science, 2006.

Grover, W. D. and Stamatelakis, D. (1998), Cycle oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration. *IEEE ICC '98, vol. 1, pp. 537-543*.

Pecorari, A. and Loiseau, I. (2012), Algorithms and Models for P-Cycle Design Without Cycle Enumeration. *Congreso Latino-Iberoamericano de Investigación Operativa, Río de Janeiro, Brasil*.

Resende, M. and Pardalos, P. (Eds.). *Handbook of Optimization in Telecommunications*. Spinger Science, 2006.

Schupke, D. A. (2004), An ILP for optimal p-cycle selection without candidate cycle enumeration. *8th Conference on Optical Network Design and Modelling ONDM'04*.

Wu, B., Yeung, K. L. and Ho, P. (2010), ILP formulations for p-Cycle design without candidate cycle enumeration. *IEEE/ACM Trans. Netw.*, v. 18, n. 1, p. 284–295.

Zhang, H. and Yang, O. Finding protection cycles in DWDM networks. *IEEE International Conference on Communications. ICC 2002.*, volume 5, p. 2756 – 2760, 2002.