Reducing Symmetry of the SONET Ring Assignment Problem using Hierarchical Inequalities

Elder M. Macambira*[1] Nelson Maculan†[1] Cid C. de Souza‡[2]

[1] Universidade Federal do Rio de Janeiro
Programa de Engenharia de Sistemas e Computação, COPPE
Caixa Postal 68511, 21941-972
e-mail: {elder,maculan}@cos.ufrj.br

[2] Universidade Estadual de Campinas
Instituto de Computação
Caixa Postal 6176, 13084-971
e-mail: cid@ic.unicamp.br

Abstract

In this paper, we consider the problem of interconnecting a set of customer sites using SONET rings of equal capacity. Each site is assigned to exactly one ring, and a special ring interconnects the other rings together. The objective is to minimize the total cost of the network subject to a ring capacity limit. This problem is called SONET Ring Assignment Problem and it can be formulated as an integer linear programming problem, which turns out to be completely symmetrical with respect to the rings. We show that the gap between the relaxed LP solution value and the optimal solution value is the largest one possible. These two properties make it very difficult to solve even smaller instances.

This paper shows that the symmetric nature and the gap can be reduced, when we impose certain decision hierarchies within the formulation. We propose two new reformulations for the SRAP. These reformulations significantly reduce the execution time of a standard commercial solver. Computational results based on these augmented formulations are presented.

Keywords: SONET Ring Assignment, Symmetry, Integer Programming, Reformulations.

1 Introduction

Most of the attention to solving any discrete optimization problem is concerned with formulation of a mathematical model, whose focus is the tightness of the underlying linear

*The author is supported by CNPq (grant 146245/1999-7) and FAPERJ (grant E-26/152.603/2001).
†correspondig author.
‡The author is supported by CNPq (grant 300883/94-3).

relaxation. However, another important feature, frequently ignored, needs special attention while formulating good integer programming models. In some circumstances, there is a natural symmetry inherent to the problem itself. Due to this property, the branch-and-bound algorithm can get difficult during the search process because the solution tree will grow exponentially, since an intermediate LP solution corresponding to a node in the tree will be repeated for all possible combinations.

One solution for the drawback presented above consists of augmenting the model with symmetry-breaking hierarchical constraints. It turns out that this improves the possibility of solving the model by a branch-and-bound algorithm. Several applications have considered symmetry issues in modelling (see Holm and Sörensen [4], Meller *et al.* [6] and Sherali and Smith [8]).

Contributions In [4], Holm and Sörensen indicate that symmetry and the large gap are two factors that arise in the resolution of the graph partitioning problems. In this paper, we consider a telecommunication network design problem. This problem can be formally described as a node-partitioning problem for a given graph G. In particular, we will show how it is possible to reduce the symmetric nature of the formulation and also greatly reduce the gap, using constraint hierarchies. Furthermore, these constraints reduce, significantly, the execution time of a standard commercial solver, even when using classes of additional automatically generated valid inequalities.

Outline The paper is organized as follows. In Section 2, we present the telecommunication network design problem. In Section 3, we formulate the network design problem as an integer linear programming problem and we consider its properties. It turns out that this formulation is completely symmetrical with respect to the rings, and that the gap between the relaxed LP solution and the optimal solution is the largest one possible. In Section 4, we show how it is possible to establish certain hierarchies by constraints; thus, we are in a position to reduce the symmetric nature of the problem and the gap. In Section 5, we present the computational results for a formulation, in comparison with alternative formulations, e.g., reformulations that are augmented with suitable symmetry-breaking constraints. Finally, in Section 6, we present some conclusions and remarks about modelling and symmetry-related issues.

2 SONET Ring Assignment Problem

The typical topology of a network SONET is a collection of local rings, or simply *rings*, directly connecting a subset of *customer sites* (see Cosares *et al.* [1], Sosnosky and Wu [9] and Wasem *et al.* [10]). Each customer site sends, receives and relays messages through a device called Add-Drop Multiplexer (ADM), which requires that the ring must have a *capacity*. The capacity is limited and consists of the volume of traffic between all pairs of nodes connected by the ring. Finally, the inter-ring relations are thus possible. All rings are connected together by a special ring, called *federal ring*, through a device, the Digital Cross-connect System (DCS), which joins each ring to the federal ring. The number of DCS's required for the network design corresponds to the cost-effective of the network.

In the design of real optical networks, a set of rings has been taken into account: maximum length of a ring, maximum number of nodes in a ring and maximum traffic on a ring.

In this paper, we will exclusively deal with the problem of forming such rings. This problem requires three conditions. The first one assumes that each customer site has to be assigned to exactly one ring; the second condition assumes that the maximum capacity of each ring is bound by a common value; and finally, the third establishes that the number of rings, that is the cost of DCS's installed, has to be minimized. The problem is called SONET Ring Assignment Problem (SRAP) with equal capacity constraints. The SRAP is known to be $\mathcal{NP}$-hard (see Goldschmidt *et al.* [3]).

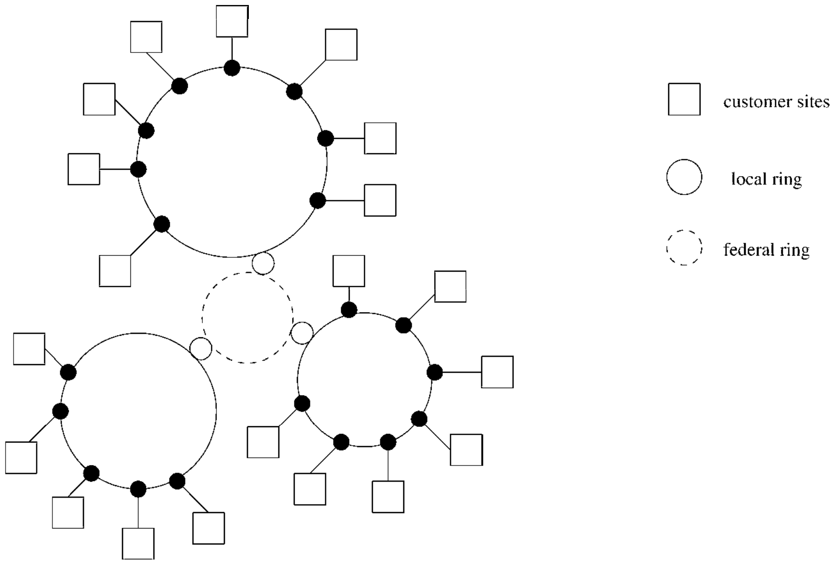

Figure 1: A SONET network.

The SONET Ring Assignment Problem can be formally described as a node-partitioning problem for a given graph G. The nodes of G represent the customer sites to be linked, and the edge weights correspond to the traffic demands among sites. Note that edge (u, v) exists only if customer sites u and v communicate. The clusters of a feasible partition to our problem are clusters of customer sites that will be placed on the same SONET ring. An example of a SONET network is given in Figure 1. In this example, we have one network with 20 sites and 3 rings.

3 Formulation and Properties of the SRAP

The Integer Programming (IP) formulation discussed here has been presented in Macambira [5]. To describe the SRAP more precisely, consider an undirected graph $G = (V, E)$ with a set V of nodes in the network indexed by $i \in V = \{1, \ldots, n\}$, where $n \geq 2$. Let d_{uv} be the traffic demand between node $u \in V$ and node $v \in V$ (that is the traffic from u to v plus the traffic from v to u). Accordingly, define an edge set $E = \{(u, v) : d_{uv} > 0\}$ comprised of such demand pairs. Furthermore, let l be an integer and let B be the capacity of a ring. Note that the common capacity B of the rings must accommodate the total demand of any ring, including the federal ring.

Then, the graph optimization problem of concern is to find a feasible partition $V_1, \ldots, V_l$

of V into disjoint clusters such that l is minimized, and the constraints about capacity of the clusters

$$\sum_{\substack{u,v \in V_i \\ u<v}} d_{uv} + \sum_{\substack{u \in V_i \\ v \notin V_i}} d_{uv} \leq B, \qquad i = 1, \ldots, l, \tag{1}$$

$$\sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \sum_{u \in V_i} \sum_{v \in V_j} d_{uv} \leq B, \tag{2}$$

may be satisfied. Note that inter-ring traffic is permissible in this design scenario and that each cluster V_k, with $k \in \{1, \ldots, l\}$, corresponds to one ring.

The integer programming formulation presented below makes use of the four group of binary variables, namely x, y, p and z. Next, we introduce these variables:

$$x_{ui} = \begin{cases} 1, & \text{if the customer site } u \text{ is assigned to ring } i, \\ 0, & \text{otherwise.} \end{cases}$$

for all $u \in V$, $i \in V$, and

$$y_i = \begin{cases} 1, & \text{if ring } i \text{ is active (that is has customer sites assigned to it),} \\ 0, & \text{otherwise.} \end{cases}$$

for all $i \in V$.

For every edge (u, v) in E, with $u < v$, and every $i \in V$, p_{uvi} is one if and only if both customer sites u and v are assigned to ring i. Finally, in the fourth group, each z variable is indexed by three elements, the first two taken from $V \times V \setminus \{(u,v) | u \in V\}$ and the last one from V.

So, according to these definitions, we have the following IP formulation for SRAP:

(P)

$$\text{maximize} \sum_{u=1}^{n}\sum_{i=1}^{n} x_{ui} - \sum_{i=1}^{n} y_i, \tag{3}$$

subject to:

$$\sum_{u=1}^{n-1}\sum_{v=u+1}^{n} d_{uv}p_{uvi} + \sum_{u=1}^{n}\sum_{v=1}^{n} d_{uv}z_{uvi} \leq B, \qquad \forall i \in V, \tag{4}$$

$$\sum_{u=1}^{n-1}\sum_{v=u+1}^{n}\sum_{i=1}^{n} d_{uv}z_{uvi} \leq B, \tag{5}$$

$$\sum_{i=1}^{n} x_{ui} \leq 1, \qquad \forall u \in V, \tag{6}$$

$$x_{ui} - y_i \leq 0, \qquad \forall u, i \in V, \tag{7}$$

$$x_{ui} + x_{vi} - p_{uvi} \leq 1, \qquad \forall u, v \in V, u < v,\ \forall i \in V, \tag{8}$$

$$x_{ui} - x_{vi} - z_{uvi} \leq 0, \qquad \forall u, v \in V,\ \forall i \in V \tag{9}$$

$$x_{ui} \in \{0, 1\}, \qquad \forall u \in V,\ \forall i \in V, \tag{10}$$

$$y_i \in \{0, 1\}, \qquad \forall i \in V, \tag{11}$$

$$p_{uvi} \in \{0, 1\}, \qquad \forall u, v \in V, u < v,\ \forall i \in V, \tag{12}$$

$$z_{uvi} \in \{0, 1\}, \qquad \forall u, v \in V,\ \forall i \in V. \tag{13}$$

Let us briefly examine the constraints in formulation (P). The objective function (3) minimizes the number of rings. Constraints (4) limit the total demand of each ring to the bandwidth B. Constraint (5) assures that the capacity of the federal ring to be less than or equal to B. Remember that the federal ring needs to carry all the traffic among nodes on different rings. Constraints (6) enforce that each customer site is assigned to at most one ring. Constraints (7) ensure that a ring is active whenever a customer site is assigned to it. For a proper triple (u, v, i), constraint (8) forces p_{uvi} to one if both customer site u and customer site v are assigned to ring i while constraint (9) forces z_{uvi} to one if customer site u is assigned to ring i and v is not assigned to that ring. Constraints (10)-(13) observe the integrality conditions on the variables x_{ui}, y_i, p_{uvi} and z_{uvi}, respectively.

In the formulation above there can be at most n rings, so that the number of y_i variables is n and x_{ui} variables is n^2. Moreover, the formulation requires variables p_{uvi} and z_{uvi} for each pair of nodes $\{u, v\}$ and one ring i. Thus, there are $2mn$ of the z_{uvi} variables and mn of the variables p_{uvi}. There are a total of $n^2 + 3mn + n$ binary variables and $n^2 + 3mn + 2n + 1$ constraints.

Model (P) is a weak formulation for two reasons. Firstly, it should be observed that the optimal value of the LP is equal to $n - 1$. This can be seen from the following feasible solution: $y_i = y_j = 1/2$, $x_{ui} = x_{uj} = 1/2$ for each $u \in V$, and all other variables are equal to zero. Thus, this solution can be considered optimal, since $\sum_{u=1}^{n}\sum_{i=1}^{n} x_{ui}$ is at most equal to n. Secondly, it should be observed that formulation (P) is completely symmetrical, because for any design, there are several equivalent designs that can be obtained by simply re-assignment of the allocations made to the various individual rings. Figure 2 shows one feasible solution S, whereas Figure 3 shows one symmetric solution to the solution S.

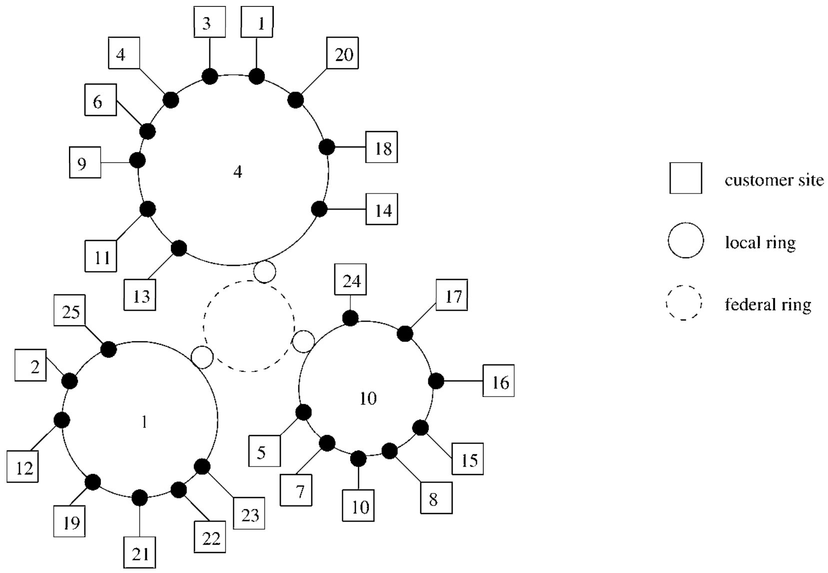

Figure 2: Example of a feasible solution.

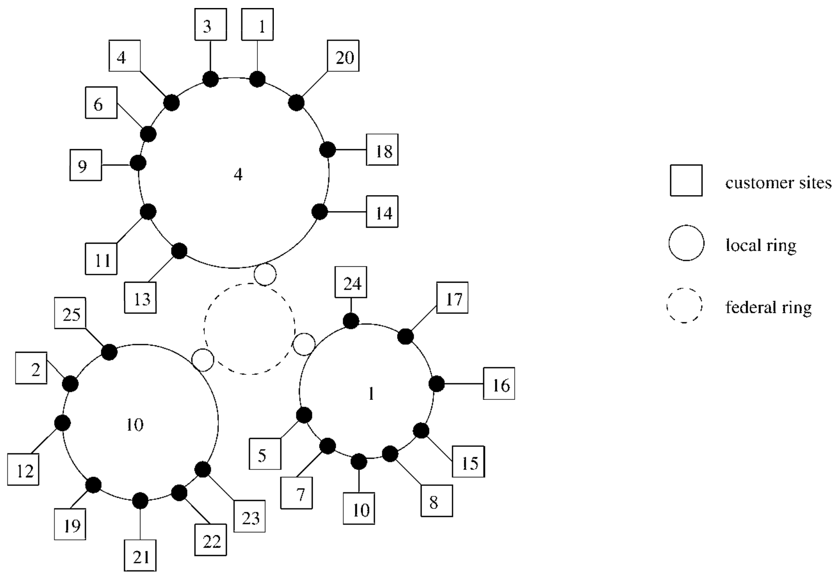

Figure 3: Example of a symmetric solution.

Let a cluster be a subset of nodes (customer sites) whose sum of demands is not greater than B, e.g. the cluster corresponds to one ring in any feasible solution. Suppose that a feasible solution consists of l cluster, with $l \leq n$. So, there are $\binom{n}{l} l!$ integer representations in the formulation (P) since the rings are indistinguishable.

Due to the difficulty described above, if the SRAP is solved by a branch-and-bound algorithm, not only will the gap be large, but the solution tree will grow exponentially, since an intermediate LP solution corresponding to a node in the tree will be repeated for all possible combinations over the rings. Hence, symmetry and the large gap are two factors that we need to overcome, if we shall be able to solve the SRAP when using exact methods. In such cases, it is beneficial to reformulate the problem by incorporating hierarchical constraints within the formulation, in order to reduce the symmetric nature and reduce the gap. In the following section, we will present some reformulations for the SONET Ring Assignment Problem.

4 Reducing the Symmetric Nature

Note that the formulation (P) has a symmetric nature, e.g., it is possible to represent many alternative optimal solutions for this formulation, because for any design, there are several equivalent designs (likewise, clones of feasible solutions). Moreover, the gap between the relaxed LP and the integer solution is the largest one possible. Thus, in this section, we attempt to break this nature and, consequently, to improve the possibility of solving the SRAP by a branch-and-bound algorithm and to lower this gap.

Reducing of the symmetric nature of the SRAP is based on adding hierarchical constraints within formulation (P), e.g., remodelling the problem. Below, we show that three constraints can be applied in order to overcome this difficulty.

The first constraint imposes a hierarchy on the number of ADM assignments, e.g., on the number of rings. Hence, the second ring is just designated if the first ring was designated, and so on. Accordingly, we propose the following constraint:

$$y_i - y_{i-1} \leq 0, \qquad i = 2, \ldots, n. \tag{14}$$

An alternative way of guaranteeing an asymmetric identity to the rings is to enforce a hierarchy based on the number of nodes (customer sites) assigned to each ring. The second constraint imposes the following:

$$\sum_{u=1}^{n} x_{ui} - \sum_{u=1}^{n} x_{u(i-1)} \leq 0, \qquad i = 2, \ldots, n. \tag{15}$$

Finally, we define the third constraint. Instead of (15), we might reduce the symmetric nature by including the following constraints:

$$\sum_{u=1}^{n} ux_{ui} - \sum_{u=1}^{n} ux_{u(i-1)} \leq 0, \qquad i = 2, \ldots, n. \tag{16}$$

In the following section, we present how the hierarchical constraints, namely, (14), (15) and (16), improve the solvability of test problems of the SONET Ring Assignment Problem.

5 Computational Results

We now describe the computational experiments that we have carried out. Our primary goal is to confirm that the hierarchical constraints we have proposed here are useful to reduce the symmetric nature of formulation (P). The second goal is to lower the gap between the optimal value of the LP and the integer solution.

To achieve our goals, we have compared the three reformulations that involve the hierarchical constraints presented in Section 4. The reformulations can be summarized as follows:

- S1: in addition to the constraints of (P), we included the constraints (14);
- S2: instead of (14), we added the constraints (15) to formulation (P);
- S3: similar to (S2), we included the constraints (16) instead of the constraints (15).

Our computational experiments were performed on a Linux PC AMD K6/450 MHz with 256 Mbytes of main memory. In these experiments, we used one class of test problems described in subsection 5.1 and, we presented more details of results in subsection 5.2.

5.1 Test Problems

We used the following class of test problems, namely, class C1, for our experiments. This class was generated by Goldschmidt *et al.* [3]. Their test problems were divided into two types of random test data:

- geometric test problems representing *natural cluster*, i.e., customer sites try to communicate more with close neighbors than with distant ones;
- random test problems were generated from complete graphs and retaining edge (u, v) with probability p, e.g., each edge of the complete graph exists in the random instance if $p \in (0, 1)$.

Each type contains both high and low demand graphs, namely the *high-demand* cases, where $622 Mbs$ ADM are being considered, and the *low-demand* cases, where the ring capacity is $155 Mbs$.

We have carried out our experiments with a set of 22 test problems: 10 high-demand and 12 low-demand. The dimension of the test problems is 15 or 25 customer sites (or nodes) with at most 52 demand pairs.

5.2 Reformulations Results

In this subsection, we present and discuss the results obtained with reformulations (S1), (S2) and (S3) for the test problems that were presented above.

For the purpose of computationally testing these alternative reformulations, the test problems have been solved by XPRESS version 12.05 (see [2]) within a maximum computing time equal to 18,000 seconds. We have also been established that the maximum number of nodes for the branch-and-bound algorithm has been set to 300,000 nodes.

In Tables 1-4, we show the results obtained with the formulation (P) and the various proposed reformulations in solving of the 22 test problems of the SRAP. In each of these tables, we have the columns: **name**: the test problem used; **# cuts**: the number of cutting planes generated via XPRESS at the root node; $\overline{z}$: the optimal value of the linear relaxation; **#nodes**: the total number of branch-and-bound nodes enumerated in the search tree by XPRESS; z^*: the optimal solution; **#rings**: the number of rings presented in the optimal solution; **time (sec.)**: the CPU time (in seconds) required to solve the test problem.

We have concentrated our analysis of the computational results on two main aspects. First, we investigate the performance in each formulation in terms of their ability in reducing symmetric nature. This is done by looking at the number of nodes opened in the enumeration tree. According to that, note that formulation (P) performs very poorly because of the number of different symmetric solutions that must be explored and eliminated by the branch-and-bound process. The reformulation (S1) outperforms reformulations (S2) and (S3), for which the number of nodes opened are greater. So, the reformulation (S1) appears to be more attractive to reduce the symmetric nature of the SRAP.

Now, if we turn our attention to the strength of the root node upper bound, the following observations apply. Note that better quality upper bounds are generated in the presence of the constraint hierarchies (14), e.g., whenever we used the reformulation (S1). In particular, note that the root node upper bound does not change when we use the reformulations (S2) and (S3). As a result, the CPU time spent in solving these test problems runs faster with reformulation (S1). In fact, the CPU time has been reduced at least 2 times whenever we used the reformulation (S1), if compared to the time obtained by formulation (P). Thus, results indicate that reformulation (S1) appears to be more attractive to lower the gap between optimal value of the LP and the integer solution.

Furthermore, the reformulations (S2) and (S3) show some promise worthy of further exploration. Apparently, because of the restrictions imposed by (15) and (16), we reduced the number of nodes enumerated in the search tree and also reduced the CPU time required as compared with formulation (P).

$B = 155Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gl_15_1	7	14	4873	3	12	654
gl_15_4	19	14	3994	3	12	665
gl_15_7	2	14	9838	3	12	1880
gl_15_9	3	14	16030	3	12	1875
rl_15_1	9	14	9118	3	12	1066
rl_15_4	4	14	5752	3	12	1201
rl_15_6	3	14	12548	3	12	1816
rl_15_8	2	14	9591	3	12	1775
rl_15_9	2	14	15270	3	12	1730
rl_15_10	9	14	8853	3	12	1825
rl_25_8	2	24	4044	3	22	2136
$B = 622Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gh_15_1	2	14	11954	3	12	2222
gh_15_2	15	14	7424	3	12	2296
gh_15_9	2	14	12357	3	12	2599
gh_25_2	2	14	923	2	13	451
rh_15_1	10	14	4361	3	12	1450
rh_15_5	2	14	9039	3	12	2378
rh_15_6	9	14	16800	3	12	3827
rh_15_9	2	14	18319	3	12	3629
rh_25_3	6	24	2354	3	12	4006
rh_25_9	2	24	353	2	23	77

Table 1: Results obtained by formulation (P).

$B = 155Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gl_15_1	18	13.40	1511	3	12	61
gl_15_4	32	13.55	1470	3	12	44
gl_15_7	13	13.00	3138	3	12	110
gl_15_9	18	13.64	441	3	12	13
rl_15_1	61	13.14	1086	3	12	14
rl_15_4	15	13.45	3432	3	12	46
rl_15_6	14	13.00	635	3	12	17
rl_15_8	26	13.07	1434	3	12	17
rl_15_9	14	13.45	1152	3	12	19
rl_15_10	16	13.55	1910	3	12	47
rl_25_8	32	13.75	3212	3	22	98
$B = 622Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gh_15_1	126	13.70	1128	3	12	131
gh_15_2	23	13.72	344	3	12	31
gh_15_9	17	13.00	888	3	12	51
gh_25_2	52	13.79	667	2	13	106
rh_15_1	6	13.00	345	3	12	28
rh_15_5	42	13.56	548	3	12	46
rh_15_6	30	13.00	1100	3	12	47
rh_15_9	23	13.62	1058	3	12	64
rh_25_3	6	13.78	192	3	12	69
rh_25_9	9	13.87	340	2	23	49

Table 2: Results obtained by reformulation (S1).

$B = 155Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gl_15_1	4	14	7359	3	12	300
gl_15_4	4	14	898	3	12	20
gl_15_7	29	14	2085	3	12	31
gl_15_9	8	14	2206	3	12	38
rl_15_1	16	14	4400	3	12	94
rl_15_4	18	14	3052	3	12	383
rl_15_6	4	14	1048	3	12	50
rl_15_8	2	14	8794	3	12	1014
rl_15_9	33	14	756	3	12	19
rl_15_10	3	14	9179	3	12	424
rl_25_8	4	24	63993	3	12	11207
$B = 622Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gh_15_1	15	14	2280	3	12	138
gh_15_2	30	14	1556	3	12	164
gh_15_9	14	14	3063	3	12	184
gh_25_2	13	24	3853	2	13	908
rh_15_1	18	14	3080	3	12	277
rh_15_5	3	14	2169	3	12	164
rh_15_6	6	14	3410	3	12	214
rh_15_9	4	14	5696	3	12	320
rh_25_3	30	24	1972	3	12	227
rh_25_9	2	24	9079	2	23	1330

Table 3: Results obtained by reformulation (S2).

$B = 155Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gl_15_1	34	14	5699	3	12	165
gl_15_4	11	14	1384	3	12	55
gl_15_7	13	14	7953	3	12	669
gl_15_9	3	14	2218	3	12	73
rl_15_1	8	14	1722	3	12	42
rl_15_4	32	14	8455	3	12	445
rl_15_6	9	14	2871	3	12	105
rl_15_8	9	14	1777	3	12	109
rl_15_9	15	14	2938	3	12	61
rl_15_10	3	14	34924	3	12	2808
rl_25_8	2	24	16254	3	22	2206
$B = 622Mbs$						
name	**#cuts**	$\overline{z}$	**#nodes**	**#rings**	z^*	**time (sec.)**
gh_15_1	8	14	2631	3	12	171
gh_15_2	5	14	1148	3	12	156
gh_15_9	2	14	4312	3	12	290
gh_25_2	1	24	10349	2	13	2786
rh_15_1	29	14	3765	3	12	367
rh_15_5	13	14	6652	3	12	540
rh_15_6	5	14	2961	3	12	183
rh_15_9	14	14	2449	3	12	142
rh_25_3	2	24	7876	3	12	3039
rh_25_9	5	24	303	2	23	43

Table 4: Results obtained by reformulation (S3).

6 Conclusions

In this paper, we have proposed three reformulations, denoted by (S1), (S2) and (S3), to reduce the effect of symmetry and to lower the gap for the SONET Ring Assignment Problem. We have run some computational experiments with a set of test problems. We have been able to show that the reformulation (S1) reduced the symmetric nature and made it possible to close the gap between the optimal value of the LP and the integer solution. Furthermore, the reformulations (S2) and (S3) outperform formulation (P) introduced earlier in the literature.

As it could be expected, the solvability of test problem of the SRAP decreases with a higher number of customer sites in the telecommunication network. This is due to a larger number of feasible partitions. Hence, future research may attempt to reinforce the study of new hierarchical constraints for reducing the symmetry inherent in the SRAP, as well as investigate automatic reformulation techniques for detecting such symmetric nature (see Rothberg [7]).

References

[1] S. Cosares, D. Deutsch, I. Saniee, and O. Wasem. SONET toolkit: a decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces*, 25(1):30–40, 1995.

[2] Dash Optimization, Inc. *XPRESS-MP: user manuals*, 1999. Version 12.05.

[3] O. Goldschmidt, A. Laugier, and E. V. Olinick. SONET/SDH ring assignment with capacity constraints. *Discrete Applied Mathematics*, 129:99–128, 2003.

[4] S. Holm and M. M. Sorensen. The optimal graph partitioning problem: solution method based on reducing symmetric nature and combinatorial cuts. *OR Spektrum*, 15:1–8, 1993.

[5] E. M. Macambira. *Modelos e Algoritmos de Programação Inteira no Projeto de Redes de Telecomunicações*. PhD thesis, Universidade Federal do Rio de Janeiro, Programa de Engenharia de Sistemas e Computação, May 2003. In portuguese.

[6] R. D. Meller, V. Narayanan, and P. H. Vance. Optimal facility layout design. *Operations Research Letters*, 23(3-5):117–127, 1998.

[7] E. Rothberg. Using cuts to remove symmetry. International Symposium on Mathematical Programming, 2000. Paper FRC316-IC107.

[8] H. D. Sherali and J. C. Smith. Improving discrete model representations via symmetry considerations. *Management Science*, 47(10):1396–1407, 2001.

[9] J. Sosnosky and Tsong ho Wu. SONET ring applications for survivable fiber loop networks. *IEEE Communications Magazine*, 29(6):51–58, June 1991.

[10] O. J. Wasem, Tsong-Ho Wu, and R. H. Cardwell. Survivable SONET networks - design methodology. *IEEE Journal on Selected Areas in Communications*, 12(1):205–212, January 1994.