

# Load Balancing of Telecommunication Networks based on Multiple Spanning Trees

Dorabella Santos\* Amaro de Sousa\* Filipe Alvelos<sup>◊</sup>

*\*Instituto de Telecomunicações  
3810-193 Aveiro, Portugal  
dorabella@av.it.pt*

*\*Instituto de Telecomunicações / DETI, Universidade de Aveiro  
3810-193 Aveiro, Portugal  
asou@ua.pt*

*◊Centro Algoritmi / DPS, Universidade do Minho  
4710-057 Braga, Portugal  
falvelos@dps.uminho.pt*

---

## Abstract

In this paper we address the problem of load balancing optimization of telecommunication networks based on multiple spanning tree routing. We focus on two objectives – minimization of the maximum link load and minimization of the network utilization imposing a worst case load value – and we propose two sets of mixed integer programming models defining the optimization problems (where one set is derived from the other). We further propose three decompositions (based on the Dantzig-Wolfe principle) to be applied to the models, as alternative means to solve the same problems. A set of computational results show the efficiency of the different models and decompositions.

**Keywords:** *multiple spanning tree protocol, integer programming, branch-and-price*

---

## 1 Introduction

This paper focuses on the load balancing optimization of telecommunication networks based on multiple spanning tree routing, which arises in the context of switched Ethernet networks. We address two objectives: the minimization of the maximum link load (maximizing the network robustness to unpredicted growth of traffic demand) and the minimization of the network utilization imposing a worst case load value (maximizing the available network resources either to accommodate new traffic demands and/or to reroute demands in the case of network failures). The aim of the second objective function is to impose the worst value given by the optimal solution of the first objective function. Therefore, to achieve the optimal solution of the second objective, we first need to determine the optimal solution of the first objective.

In switched Ethernet networks with the IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) [1], routing of traffic flows is done through the network based on spanning trees (STs). The operator defines each required ST by creating a spanning tree instance (STI) and assigning to it a set of traffic flows. When a traffic flow is assigned to a STI, it is routed between its end nodes through the unique path defined by the assigned STI. MSTP also enables the set of STIs over a set of regions defined on the network (it ensures that failures inside a region do not affect routing outside that region minimizing

the impact of network failures). Nevertheless, it has been shown in a previous work of one the authors on this paper [2] that, with optimized configurations, considering the whole network as a single region provides the best results with respect to load balance and service disruption and, here, we consider the single region approach. MSTP has been recently proposed to enhance traffic engineering capabilities of Ethernet networks [3, 4, 5, 6]. Nevertheless, as far as we are aware, there is no reference dealing with exact methods to find optimal solutions for the load balancing objectives addressed here in the context of multiple spanning tree routing based networks.

In this paper, we first propose in Section 2 two sets of mixed integer linear programming models defining the two objectives: a first natural set and a second which is a reformulation of the first one. Then, in Section 3, we describe different reformulations based on the Dantzig-Wolfe decomposition, which are solved by branch-and-price, as alternative means to solve the optimization problems. Finally, we present in Section 4 the computational results showing the efficiency of the different models and decompositions.

## 2 Compact Models

Consider a network defined on a graph  $G(N, A)$  where  $N$  is the set of nodes and  $A$  is the set of links between nodes. Edge  $\{i, j\} \in A$  represents a link between node  $i \in N$  and  $j \in N$ , while arc  $(i, j) \in A$  represents the edge  $\{i, j\}$  directed from  $i$  to  $j$ . Let  $V(i)$  be the set of neighbor nodes of  $i \in N$  on graph  $G(N, A)$ . Each edge has a known capacity given by  $c_{\{ij\}} (= c_{ij} = c_{ji})$ . The network is to support a set of commodities  $K$  where each commodity  $k \in K$  is characterized by its origin node  $o_k \in N$ , its destination node  $d_k \in N$  and its demand  $b_k \in \mathbb{R}^+$ . Commodities must be routed over the network on paths defined by one of a set of spanning trees  $S$  of size  $|S|$ . The aim is to choose set  $S$  and to assign the commodities to the spanning trees.

### 2.1 First Set of Compact Models

Let  $r$  be one node of  $N$  and referred to as “root node”. The basic decision variables are:

$$\begin{aligned} x_{ij}^k &\in \{0, 1\} - \text{indicates if arc } (i, j) \in A \text{ is in the path of the commodity } k \in K \\ \phi_k^s &\in \{0, 1\} - \text{indicates if commodity } k \in K \text{ is assigned to spanning tree } s \in S \\ \beta_{\{ij\}}^s &\in \{0, 1\} - \text{indicates if edge } \{i, j\} \in A \text{ is in the spanning tree } s \in S \end{aligned}$$

We need also the following variables in order to model the spanning tree constraints:

$$\begin{aligned} y_{ij}^{zs} &\in \{0, 1\} - \text{indicates if arc } (i, j) \in A \text{ is in the path from node } z \in N \setminus \{r\} \text{ to the root node } r \text{ on} \\ &\text{the spanning tree } s \in S \\ \theta_{ij}^s &\in \{0, 1\} - \text{indicates if arc } (i, j) \in A \text{ is in the spanning tree } s \in S \end{aligned}$$

All models have to be compliant with a set of common constraints. The first set is:

$$\sum_{j \in V(i)} (y_{ij}^{zs} - y_{ji}^{zs}) = \begin{cases} 1, & i = z \\ 0, & i \neq z \end{cases} \quad \forall z \in N \setminus \{r\}, \forall s \in S, \forall i \in N \setminus \{r\} \quad (1)$$

$$\theta_{ij}^s \geq y_{ij}^{zs} \quad \forall z \in N \setminus \{r\}, \forall (i, j) \in A, \forall s \in S \quad (2)$$

$$\sum_{j \in V(i)} \theta_{ij}^s = \begin{cases} 1, & i \neq r \\ 0, & i = r \end{cases} \quad \forall i \in N, \forall s \in S \quad (3)$$

$$\theta_{ij}^s + \theta_{ji}^s = \beta_{\{ij\}}^s \quad \forall \{i, j\} \in A, \forall s \in S \quad (4)$$

which guarantees that the set of variables  $\beta_{\{ij\}}^s$  define the spanning trees. For each  $s \in S$ : constraints (1) guarantee that variables  $y_{ij}^{zs}$  define a path from  $z \in N$  to root node  $r \in N$ ; constraints (2) guarantee that variables  $\theta_{ij}^s$  include the arcs of all paths defined by variables  $y_{ij}^{zs}$ ; constraints (3), together with constraints (2), guarantee that the paths given by variables  $y_{ij}^{zs}$  define a directed spanning tree towards the root node  $r$ ; constraints (4) guarantee that the variables  $\beta_{\{ij\}}^s$  include the edges of the directed

spanning tree defined by variables  $\theta_{ij}^s$ . Note that the equalities (4) could be used to eliminate variables  $\beta_{\{ij\}}^s$  from the model but they are useful to define the decompositions. The second set is:

$$x_{ij}^k + x_{ji}^k \leq \beta_{\{ij\}}^s + (1 - \phi_k^s) \quad \forall \{i, j\} \in A, \forall k \in K, \forall s \in S \quad (5)$$

$$\sum_{j \in V(i)} (x_{ij}^k - x_{ji}^k) = \begin{cases} 1, & i = o_k \\ 0, & i \neq o_k, d_k \\ -1, & i = d_k \end{cases} \quad \forall i \in N, \forall k \in K \quad (6)$$

where constraints (6) are the path conservation constraints which together with (5) guarantee that the path is defined on links belonging to the assigned spanning tree (when variable  $\phi_k^s$  is 1, constraints (5) impose that  $x_{ij}^k + x_{ji}^k \leq \beta_{\{ij\}}^s$  and when variable  $\phi_k^s$  is 0, constraints (5) are redundant). The third set is:

$$\sum_{s \in S} \phi_k^s = 1 \quad \forall k \in K \quad (7)$$

which guarantees that each commodity is assigned to one spanning tree.

For the first objective function (the minimization of the maximum link load), we use the decision variable  $\mu \in [0, 1]$  to account for the maximum load of any edge and the optimization problem is given by the following mixed integer linear programming (MILP) model:

$$\begin{aligned} & \min \mu \\ & \text{s.t.} \\ & (1) - (7) \\ & \sum_{k \in K} b_k (x_{ij}^k + x_{ji}^k) \leq c_{\{ij\}} \mu \quad \forall \{i, j\} \in A \\ & y_{ij}^{zs} \in \{0, 1\}, \theta_{ij}^s \in \{0, 1\}, \phi_k^s \in \{0, 1\}, x_{ij}^k \in \{0, 1\}, \beta_{\{ij\}}^s \in \{0, 1\}, \mu \in [0, 1] \end{aligned} \quad (8)$$

For the second objective function (the minimization of the network utilization imposing a worst case load value  $\mu_{\max}$ ), the decision variables are  $\mu_{\{ij\}}$  which account for the load of each edge  $\{i, j\} \in A$  and the MILP model defining this optimization problem is given by:

$$\begin{aligned} & \min \sum_{\{i, j\} \in A} \mu_{\{ij\}} \\ & \text{s.t.} \\ & (1) - (7) \\ & \sum_{k \in K} b_k (x_{ij}^k + x_{ji}^k) = c_{\{ij\}} \cdot \mu_{\max} \cdot \mu_{\{ij\}} \quad \forall \{i, j\} \in A \\ & y_{ij}^{zs} \in \{0, 1\}, \theta_{ij}^s \in \{0, 1\}, \phi_k^s \in \{0, 1\}, x_{ij}^k \in \{0, 1\}, \beta_{\{ij\}}^s \in \{0, 1\}, \mu_{\{ij\}} \in [0, 1] \end{aligned} \quad (9)$$

## 2.2 Second Set of Compact Models

Alternative MILP models defining the same optimization problems are obtained by disaggregating each variable  $x_{ij}^k$  into variables  $x_{ij}^{ks}$ , one for each  $s \in S$ . In this case, variable  $x_{ij}^{ks}$  indicates if arc  $(i, j) \in A$  is in the path of commodity  $k \in K$  in the assigned spanning tree  $s \in S$ .

With this disaggregation technique, the common constraints (5)-(6) are reformulated as:

$$x_{ij}^{ks} + x_{ji}^{ks} \leq \beta_{\{ij\}}^s \quad \forall \{i, j\} \in A, \forall k \in K, \forall s \in S \quad (10)$$

$$\sum_{j \in V(i)} (x_{ij}^{ks} - x_{ji}^{ks}) = \begin{cases} \phi_k^s, & i = o_k \\ 0, & i \neq o_k, d_k \\ -\phi_k^s, & i = d_k \end{cases} \quad \forall i \in N, \forall k \in K, \forall s \in S \quad (11)$$

Constraints (11) are straightforward under the proposed variable disaggregation. Concerning constraints (10), the commodity to spanning tree assignment information, which is defined in (5) by variables  $\phi_k^s$ , is now defined by variables  $x_{ij}^{ks}$  and the term  $(1 - \phi_k^s)$  of constraints (5) is no longer required on (10).

It is possible to prove that the first set, although more compact than the second one (it has less variables and less constraints), has a convex hull that accommodates solutions which do not belong to that of the second set. Consider the linear relaxation of both sets assuming, for example, that the spanning tree constraints define two integer spanning trees,  $s = 1$  and  $s = 2$  and that  $\phi_k^1 = \phi_k^2 = 0.5$  for some  $k \in K$ . Note that in the second set, under these conditions, constraints (10) and (11) guarantee that 50% of the demand for commodity  $k$  is routed through tree  $s = 1$ , while the other 50% is routed through the other tree,  $s = 2$ . However, in the first set, the right end side of constraints (5) is always at least 0.5, which means that 50% of the demand can be freely routed by the constraint (5) of spanning tree  $s = 1$  and the other 50% can be also freely routed by the constraint (5) of spanning tree  $s = 2$ , proving the statement that the convex hull of the first set accommodates solutions which do not belong to that of the second set. Computational results will show which one leads to better performance.

The resulting MILP model for the minimization of the maximum link load objective is given by:

$$\begin{aligned}
& \min \mu \\
& \text{s.t.} \\
& (1) - (4), (7), (10) - (11) \\
& \sum_{s \in S} \sum_{k \in K} b_k (x_{ij}^{ks} + x_{ji}^{ks}) \leq c_{\{ij\}} \mu \quad \forall \{i, j\} \in A \\
& y_{ij}^{zs} \in \{0, 1\}, \theta_{ij}^s \in \{0, 1\}, \phi_k^s \in \{0, 1\}, x_{ij}^{ks} \in \{0, 1\}, \beta_{\{ij\}}^s \in \{0, 1\}, \mu \in [0, 1]
\end{aligned} \tag{12}$$

and for the minimization of the network utilization imposing a worst case load value  $\mu_{\max}$  is given by:

$$\begin{aligned}
& \min \sum_{\{i, j\} \in A} \mu_{\{ij\}} \\
& \text{s.t.} \\
& (1) - (4), (7), (10) - (11) \\
& \sum_{s \in S} \sum_{k \in K} b_k (x_{ij}^{ks} + x_{ji}^{ks}) = c_{\{ij\}} \cdot \mu_{\max} \cdot \mu_{\{ij\}} \quad \forall \{i, j\} \in A \\
& y_{ij}^{zs} \in \{0, 1\}, \theta_{ij}^s \in \{0, 1\}, \phi_k^s \in \{0, 1\}, x_{ij}^{ks} \in \{0, 1\}, \beta_{\{ij\}}^s \in \{0, 1\}, \mu_{\{ij\}} \in [0, 1]
\end{aligned} \tag{13}$$

### 3 Decompositions

This section briefly describes different possible decompositions, based on the Dantzig-Wolfe principle [7], which we have solved by branch-and-price (B&P) [8], as alternative means to solve the optimization problems. In our case, the branch constraints are added to the master problem and are defined using the original variables (the ones from the compact model).

The first decomposition considers the set of spanning tree constraints, i.e., constraints (1)-(4), as the subproblems (one for each required spanning tree). These subproblems define the minimum cost spanning tree problem which we efficiently solve using the Kruskal algorithm.

The second decomposition considers the path constraints as subproblems. For the first set of models, constraints (6) define the subproblems (one for each required commodity). The optimal solution of each subproblem is a path. On the other hand, for the second set of models, constraints (11) are the subproblems (one for each commodity and for each required spanning tree). Note that, in this case, the optimal solution of a subproblem can be either a path (with variables  $x_{ij}^{ks}$  defining the path and variable  $\phi_k^s$  equal to 1) or a non-path (all variables  $x_{ij}^{ks}$  equal to 0 and variable  $\phi_k^s$  also equal to 0). These subproblems can be efficiently solved using the Bellman-Ford algorithm.

The third decomposition is the combination of the two previous ones. In this case, the subproblems are either minimum cost spanning tree problems or shortest path problems.

## 4 Computational Results

Computational results for the compact formulations, as well as for the decompositions, were obtained with the networks depicted in Figure 1 on a Pentium IV machine at 3.4 GHz and with 1GB of RAM. In all cases, we have considered a number of spanning trees  $|S| = 2$  and a maximum runtime of 2 hours was allowed. Two sets of flow demands were considered for each network – Set I and II for Network A – and Set III and IV for Network B – where Set I and III are significantly more unbalanced (in terms of total traffic per network node) than Set II and IV, respectively.

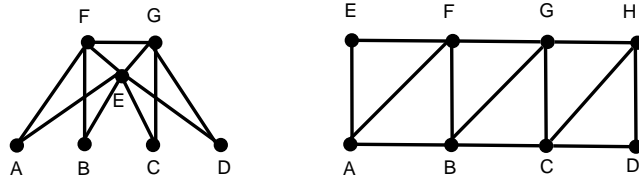


Figure 1: Network A with 7 nodes and 11 links (left) and Network B with 8 nodes and 13 links (right).

The compact formulations were solved using CPLEX 9.1. The  $\mu_{\max}$  value given to the second objective function was always the value of the optimal solution of the first objective function. The results for the first set of compact models are shown in Table 1, whereas the results for the second compact model are shown in Table 2: the results include the LP value (value of the objective function for the Linear Programming relaxation), the MIP value (the value of the objective function of the optimal integer solution), the total computational time in seconds and the number of nodes of the branch-and-bound (B&B) search tree.

	Min of max link load				Min of network utilization imposing a worst case value			
	LP	MIP	Time	BB	LP	MIP	Time	BB
A & I	0.525	0.55	70.67	39434	3.35	3.45	0.08	1
A & II	0.325	0.35	55.05	18998	2.65	2.65	0.03	1
B & III	0.467	0.50	3069.78	766878	3.65	3.65	0.14	1
B & IV	0.533	0.55	3331.03	564598	4.45	4.50	7.49	1454

Table 1: Computational results for the first set of compact models

	Min of max link load				Min of network utilization imposing a worst case value			
	LP	MIP	Time	BB	LP	MIP	Time	BB
A & I	0.525	0.55	51.6	16354	3.35	3.45	4.08	1384
A & II	0.325	0.35	22.6	4992	2.65	2.65	0.03	1
B & III	0.467	0.50	450.4	81887	3.65	3.65	2.53	161
B & IV	0.533	0.55	738.7	100952	4.45	4.50	8.89	1205

Table 2: Computational results for the second set of compact models

First of all, these results show that the models of the first objective function are much harder to solve than the ones of the second objective function. Then, the disaggregation technique used in the second set of models improves CPLEX efficiency in the case of the first objective function (huge reduction of B&B nodes and computational time) but it has worse performance in the second case (the first model finds the optimal integer solution at the very first LP node in three out of four instances, while this happens only for one instance in the second model and all computational times are better for the first model).

The proposed decompositions were implemented with the ADDing (Automatic Dantzig-Wolfe Decomposition for Integer Column Generation) set of C++ classes [9], where the restricted master problems were solved using CPLEX 9.1. In the implementation of the B&B tree search, we have implemented different node and variable selection strategies. Concerning the node selection strategy, we have considered: depth-first, breadth-first and best-bound. In average, depth-first and best-bound showed similar performance, while breadth-first was worst (the computational results shown here refer to the results with depth-first). Concerning variable selection strategy, we have tested a significant number of different strategies. In the overall, the best strategy was to choose: (i) closest value to 1 among variables  $\beta_{\{i,j\}}^s$  of most loaded links; (ii) if all these variables are integer, then closest value to 1 among the remaining variables  $\beta_{\{i,j\}}^s$ ; (iii) if all variables  $\beta_{\{i,j\}}^s$  are integer, then closest value to 1 among variables  $\phi_k^s$  of commodities using most loaded links; (iv) if all these variables are integer, then closest value to 1, among the remaining variables  $\phi_k^s$ .

Concerning the minimization of maximum link load, Tables 3 and 4 present the computational results of all three decompositions applied to the first and second compact models, respectively. Each table presents separately the number of B&B nodes and the computational times of all decompositions (in the first case, we repeat the B&B nodes of the CPLEX runs for comparison reasons).

	Min of max link load				Min of network utilization imposing a worst case value			
	CPX	Tree	Path	Tree+Path	CPX	Tree	Path	Tree+Path
A & I	39434	38709	482119	77422	1	858	159	9
A & II	18998	14365	184439	49988	1	5	1	3
B & III	766878	122186*	207139*	256036*	1	140	1	2971
B & IV	564598	125117*	174373*	232282*	1453	24121	32410	13563

	Min of max link load			Min of network utilization imposing a worst case value		
	Tree	Path	Tree+Path	Tree	Path	Tree+Path
A & I	399.64	5224.80	543.80	10.38	1.05	0.2
A & II	138.08	2323.73	369.30	0.17	0.17	0.16
B & III	7200	7200	7200	3.36	2.34	51.56
B & IV	7200	7200	7200	1161.28	612.55	344.84

\*Reached time limit of 7200 seconds

Table 3: Computational results of decompositions applied to the first set of compact models.

Let us consider the first objective function. In the case of the first compact model (Table 3), and for the instances that were solved within the given maximum runtime limit, the tree decomposition has the best performance (both in terms of B&B nodes and in terms of computational time) and the path decomposition has the worst performance. Moreover, the tree decomposition is slightly better than CPLEX in number of B&B nodes (the computational times are not comparable since CPLEX is an highly efficient commercial software package and the branch-and-price approaches were implemented by the authors). In the case of the second compact model (Table 4), the path decomposition becomes much more efficient and, therefore, the best decomposition. Note that although the path decomposition generates on average a little less number of B&B nodes than the tree decomposition, its merit is more

	Min of max link load				Min of network utilization imposing a worst case value			
	CPX	Tree	Path	Tree+Path	CPX	Tree	Path	Tree+Path
A & I	16354	24461	17571	14222	1384	552	313	431
A & II	4992	4879	7265	5129	1	3	1	4
B & III	81887	79543	70677	65928	161	43	1	72
B & IV	100952	35654	27291	39266	1205	804	1057	872

	Min of max link load			Min of network utilization imposing a worst case value		
	Tree	Path	Tree+Path	Tree	Path	Tree+Path
A & I	429.25	150.53	155.67	10.47	2.25	3.78
A & II	113.17	71.61	75.17	0.22	0.16	0.16
B & III	4972.30	1247.47	2954.69	1.66	1.31	1.34
B & IV	3673.50	755.83	2375.39	31.97	26.80	66.86

Table 4: Computational results of the decompositions applied to the second set of compact models.

apparent in its much shorter computational times. Finally, we see in both cases that combining tree and path decompositions is not worthy since its performance is, in all cases, between the performance of each individual decomposition.

Let us now consider the second objective function. In the case of the first compact model (Table 3), the path decomposition was in general the best approach but the combined tree and path decomposition was the best in two cases. Nevertheless, all decompositions performed worse than the compact model solved by CPLEX. In the case of the second compact model (Table 4), the path decomposition was the best approach and all decompositions performed better than the compact model solved by CPLEX (in terms of generated B&B nodes). In the overall, the performance of the decompositions is better when applied to the second compact model, in particular in the most difficult problem instances. Moreover, unlike the first objective function, in this case there are cases where the combined tree and path decomposition performs better than the individual decompositions.

## 5 Conclusion

Here, we have addressed two load balancing related objective functions applied to telecommunication networks based on multiple spanning tree routing. Two sets of mixed integer linear programming models defining the optimization problems were proposed and compared. The second model set, which is derived from the first one based on a variable disaggregation technique, was found to be much more efficient for the minimization of the maximum link load but less efficient for the minimization of the network utilization imposing a worst case load value. Three different decompositions that can be applied to both model sets were also proposed and compared. In the overall, the path decomposition applied to the second model set is the best technique amongst all.

We note that the proposed decompositions do not improve the lower bound of the restricted master problem, since the respective subproblems have the integrality property, i.e, the subproblems have integer optimal solutions. In an attempt to improve the lower bound, we are investigating other decompositions, namely decompositions involving constraints (1)-(4) and (10)-(11) in the subproblems and certain variants of the path decomposition, although efficient algorithms for some of these subproblems are not known.

We note also that techniques based on disaggregation of variables have been also used in other contexts like, for example, in capacitated network design problems (disaggregating the decision variables on the number of facilities, see [10]) or hop constrained network design problems (disaggregating the routing variables in their position from the origin node, see [11]). These techniques sometimes allow the

identification of additional inequalities that improve lower bounds. Thus, we are in the process of identifying such inequalities adequate to the problems addressed here and investigating if they can improve the efficiency of the solution techniques through row generation techniques.

## Acknowledgment

The authors would like to thank the portuguese FCT (Fundação para a Ciência e a Tecnologia) for its support through projects PTDC/EIA/64772/2006 and POSC/EIA/57203/2004 and through the post-doc grant SFRH/BPD/41581/2007 of the first author.

## References

- [1] IEEE Standard 802.1s, “Virtual Bridged Local Area Networks - Amendment 3: Multiple Spanning Trees”, 2002
- [2] A. F. de Sousa, G. Soares, “Improving Load Balance and Minimizing Service Disruption on Ethernet Networks using IEEE 802.1S MSTP ”, EuroFGI Workshop on IP QoS and Traffic Control, IST Press, pp. 25 - 35, 2007
- [3] A. Kern, I. Moldovan, T. Cinkler, “Scalable Tree Optimization for QoS Ethernet”, IEEE Symp. on Computers and Communications (ISCC’06), pp. 578 - 584, 2006
- [4] M. Ali, G. Chiruvolu, A. Ge, “Traffic Engineering in Metro Ethernet”, IEEE Network, Vol. 19, No. 2, pp. 10 - 17, 2005
- [5] A. Kolarov, B. Sengupta, A. Iwata, “Design of Multiple Reverse Spanning Trees in Next Generation of Ethernet-VPNs”, IEEE GLOBECOM’04, Vol. 3, pp. 1390 - 1395, 2004
- [6] S. Sharma, K. Gopalan, S. Nanda, T. Chiueh, “Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks”, IEEE INFOCOM’04, Vol. 4, pp. 2283 - 2294, 2004
- [7] G. B. Dantzig and P. Wolfe, “Decomposition principle for linear programs”, Operations Research, Vol 8, pp. 101-111, 1960
- [8] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, “Branch-and-price: Column generation for solving huge integer programs”, Operations Research, Vol. 46, pp. 316-329, 1998
- [9] F. Alvelos, “Branch-and-price and multicommodity flows”, PhD Thesis, Universidade do Minho, 2005
- [10] A. Frangioni, B. Gendron, “0-1 reformulations of multicommodity capacitated network design problem”, Discrete Applied Mathematics, doi:10.1016/j.dam.2008.04.022, 2008
- [11] L. Gouveia L, P. Patrício, A.F. de Sousa, “Hop-Constrained Node Survivable Network Design: An Application to MPLS over WDM ”, Networks and Spatial Economics, Vol. 8, No. 1, pp. 3 - 21, 2008