Instituto de Engenharia de Sistemas e Computadores de Coimbra
Institute of Systems Engineering and Computers
INESC – Coimbra

M. Eugénia Captivo, João Clímaco and Marta Pascoal

A mixed integer linear formulation for the minimum label spanning tree problem

No. 11 2008

ISSN: 1645-2631

Instituto de Engenharia de Sistemas e Computadores de Coimbra
INESC – Coimbra
Rua Antero de Quental, 199; 3000 - 033 Coimbra; Portugal
www.inescc.pt

A mixed integer linear formulation for the minimum label spanning tree problem

M. Eugénia Captivo[1], João C. N. Clímaco[2,3], Marta M. B. Pascoal[2,4]

[1] Faculdade de Ciências, Universidade de Lisboa
Centro de Investigação Operacional
Campo Grande, Bloco C6, 1749-016 Lisboa, Portugal
E-mail: mecaptivo@fc.ul.pt

[2] Instituto de Engenharia de Sistemas e Computadores – Coimbra
Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

[3] Faculdade de Economia da Universidade de Coimbra
Avenida Dias da Silva, 165, 3004-512 Coimbra, Portugal
E-mail: jclimaco@inescc.pt

[4] Departamento de Matemática da Universidade de Coimbra,
Apartado 3008, 3001-454 Coimbra, Portugal
Phone: +351 239 791150, Fax: +351 239 832568
E-mail: marta@mat.uc.pt

August 2008

Abstract: In this paper we deal with the minimum label spanning tree problem. This is a relevant problem with applications such as telecommunication networks or electric networks, where each edge is assigned with a label (such as a color) and it is intended to determine a spanning tree with the minimum number of different labels. We introduce some mixed integer formulations for this problem and prove that one of their relaxation always gives the optimal value. Finally we present and discuss the results of computational experiments.

Keywords: Spanning tree, Label, Color, Mixed integer formulation.

1 Introduction

The determination of spanning trees leads to several optimization problems with many applications, specially when the network connectivity is a requirement. One of the most studied of those problems is the minimum cost spanning tree problem (or simply minimum spanning tree problem), the goal of which is to find a minimum cost connected subgraph of a network that optimises an additive objective function. This problem can be solved in polynomial time using, for instance, the algorithms proposed by Kruskal [5], in 1956, or by Prim [6], in 1957. However, other objective functions are of interest. One of those cases, and also a more recent one, being the minimum label spanning tree problem (MLSTP), with the goal of determining the most uniform subgraph of a network, assuming each edge is associated with a label (or color). This type of problem has many applications in telecommunications, as different colors can be seen as different operators or different types of calls. The problem was introduced in 1997 by Chang & Leu [2], who proved its NP-hardness, by reducing it to a minimum cover problem. Despite this problem being harder than the minimum cost spanning tree problem, Chang & Leu developed an exact exponential algorithm, as well as two heuristic methods that perform $\mathcal{O}(mn)$ and $\mathcal{O}(\ell mn)$ operations, where ℓ, n and m are the number of distinct colors, vertices and edges in the network, respectively. Since then other researchers have studied and presented other heuristics for the MLSTP. Some references on this topic are [1, 3, 4, 7, 8, 9].

The manuscript is organised as follows. Section 2 introduces the notation and preliminary concepts, proposes some mixed integer linear formulations for the MLSTP and proves that one of their relaxation gives the optimal value. Section 3 reports the results of computational experiments.

2 The minimum labeling spanning tree problem

In the following we consider an undirected network $(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ denotes the set of n vertices and $\mathcal{A}$ the set of m edges. Let $\mathcal{L} = \{1, \ldots, \ell\}$ be a set of labels (or colors) and assume each edge $\{i, j\}$ is associated with a value $l_{ij} \in \mathcal{L}$. Given a vertex i in $\mathcal{N}$, $\Gamma^-(i)$ will represent the set of edges (or arcs) that end at i and $\Gamma^+(i)$ will represent the set of edges (or arcs) that start at i.

For each spanning tree T of $(\mathcal{N}, \mathcal{A})$, the number of distinct colors of its edges is denoted by $l(T)$. Giving each edge a label, the MLSTP is to find a spanning tree whose edges are as similar as possible, that is, that use the smallest possible number of different labels. The goal is then to find a spanning tree T that minimizes $l(T)$.

To formulate the problem of determining the minimum labeling spanning tree in Mixed Integer Linear Programming, we consider the rooted oriented single commodity flow model. The root will be vertex 1 and each edge $\{i, j\} \in \mathcal{A}$ with $i \neq 1$ and $j \neq 1$, originates two arcs (i, j) and (j, i). Edges $\{1, j\}$ or $\{j, 1\}$ only originate arc $(1, j)$, because we want to determine the flow from the root to all the other vertices in $\mathcal{N}$ minimizing some objective function. We denote by $\mathcal{A}'$ the set of arcs obtained, in this way, from the edge set $\mathcal{A}$. The subset of $\mathcal{A}$ $(\mathcal{A}')$ formed by the edges (arcs) with label l will be denoted by $\mathcal{A}_l$ $(\mathcal{A}'_l)$.

For each arc $(i, j) \in \mathcal{A}'$, we consider:

- the topological variables

$$X_{ij} = \begin{cases} 1 \text{ if arc } (i,j) \text{ is in the spanning tree} \\ 0 \text{ otherwise} \end{cases}.$$

- the flow variables f_{ij} denoting the flow trough arc (i, j).

Knowing that each arc (i, j) has a label l, we also need to consider, for each $l \in \mathcal{L}$, variables

$$v_l = \begin{cases} 1 \text{ if label } l \text{ is in some arc of the spanning tree} \\ 0 \text{ otherwise} \end{cases}.$$

Then, our model for the minimum labeling spanning tree problem, (MLST), is:

$$\begin{aligned}
\min \quad & \sum_{l \in \mathcal{L}} v_l & & \\
\text{subject to} \quad & \sum_{i \in \Gamma^-(j)} X_{ij} = 1, & j \in \mathcal{N} - \{1\} & \quad (1) \\
& \sum_{i \in \Gamma^-(j)} f_{ij} - \sum_{i \in \Gamma^+(j)} f_{ji} = 1, & j \in \mathcal{N} - \{1\} & \quad (2) \\
& X_{ij} \leq f_{ij} \leq (n-1) X_{ij}, & (i,j) \in \mathcal{A}' & \quad (3) \\
& \sum_{(i,j) \in \mathcal{A}'_l} X_{ij} \leq \min\{n-1, |\mathcal{A}_l|\} v_l, & l \in \mathcal{L} & \quad (4) \\
& X_{ij} \in \{0, 1\}, & (i,j) \in \mathcal{A}' & \quad (5) \\
& v_l \in \{0, 1\}, & l \in \mathcal{L} & \quad (6) \\
& f_{ij} \geq 0, & (i,j) \in \mathcal{A}' & \quad (7)
\end{aligned}$$

Constraints (1) assure that the spanning tree (solution) has exactly one arc incident in each vertex. Constraints (2) guarantee flow conservation for all vertices different from the root leaving one unit of flow at each vertex. Constraints (3) imply that the flow in each arc can only be positive if the arc belongs to the spanning tree, and it can not exceed the number of vertices minus 1. Constraints (4) assure that for each label, an arc with that label can only be in the solution if the associated label variable is equal to 1. It also implies that the number of arcs in the solution with each label cannot exceed the number of vertices minus 1, neither the number of edges with that label in the original network. Constraints (5) and (6) are the integrality constraints for X_{ij} and v_l. Constraints (7) assure that the flow variables are always non-negative.

In [8] we found a Mixed Integer Linear Programming formulation for a related problem, the label-constrained minimum spanning tree problem[1] (LC-MSTP), made by Si Chen. The topological variables used are X_{ij}^k, with

$$X_{ij}^k = \begin{cases} 1 & \text{if edge } (i,j) \text{ and label } k \text{ are used in the spanning tree} \\ 0 & \text{otherwise} \end{cases}.$$

Quite recently [10] presented another Mixed Integer Linear Programming formulation for the LC-MSTP, where the topological variables used are e_{ij} with

$$e_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is used} \\ 0 & \text{otherwise} \end{cases}.$$

As it was shown in [2], the MLSTP is NP-hard. So, instead of solving the model (MLST) using Mixed Integer Linear Programming we can obtain a lower bound by solving its linear relaxation. We just need to use

$$0 \le X_{ij} \le 1, \ (i,j) \in \mathcal{A}' \tag{8}$$

and

$$0 \le v_l \le 1, \ l \in \mathcal{L} \tag{9}$$

instead of (5) and (6). We can also impose the disaggregated constraints:

$$X_{ij} \le v_l, \ l \in \mathcal{L}, \ (i,j) \in \mathcal{A}'_l \tag{10}$$

Including constraints (10) allows for a better bound when solving the linear relaxation of (MLST). We will denote this model by (MLSTa) (including constraints (1) to (7) and (10)).

Let $v(P)$ denote the optimal value of problem P and $v_L(P)$ denote the optimal value of the linear relaxation of problem P. As it is well known and our computational experience confirms, $v_L(\text{MLSTa})$ is always greater than or equal to $v_L(\text{MLST})$ and the difference may be significant. We also solved in Mixed Integer Linear Programming the relaxation of the model (MLST) obtained by including constraints (1) to (4), (6) to (8) and (10), which we will denote by (MLSTb). In this case, only variables v_l are restricted to have 0 or 1 values. In all instances solved we obtained $v(\text{MLSTb}) = v(\text{MLST})$. The same happens if we solve in Mixed Integer Linear Programming the relaxation of the model (MLST) obtained by including constraints (1) to (4) and (6) to (8), which we will denote by (MLSTc). Again in this case, only variables v_l are restricted to have 0 or 1 values. We can prove that this will always be the case, i.e.:

$$v(\text{MLSTb}) = v(\text{MLSTc}) = v(\text{MLST}).$$

We know that

$$v(\text{MLSTb}) \le v(\text{MLST}) \text{ and } v(\text{MLSTc}) \le v(\text{MLST}),$$

[1]The LC-MSTP aims to compute the minimum cost spanning tree with at most K labels, for a given positive integer K.

since (MLSTb) and (MLSTc) are relaxations of (MLST). To show that the equality holds we just have to build a feasible integer solution to (MLST) using the same number of labels of the optimal solution of v(MLSTb) or the optimal solution of v(MLSTc). If the optimal solution to (MLSTb) and (MLSTc) is integer it is done. If the optimal solution to any of these two problems is not all integer, we can build an integer solution with the same objective value, by applying the following procedure:

- Start with $S = \{1\}$ and $R = N - \{1\}$;
- Repeat until R is empty:
  - For each vertex $i \in S$, if there exists some variable X_{ij} ($j \in R$) with positive value at the optimal solution of (MLSTb) (or (MLSTc)) we put $Y_{ij} = 1$, remove vertex j from R and insert it in S.

In the optimal solution of any of the models (MLSTb) (or (MLSTc)), one unit of flow is going from the root to each vertex. So, there is at least one path from the root (vertex 1) to each vertex in $\mathcal{N} - \{1\}$. If the solution is all integer it is exactly one path and all the arcs in the path have $X_{ij} = 1$. If the solution is not all integer, then more than one path is used, each one with value smaller than 1. Our procedure arbitrarily chooses one of those paths, and gives it the value 1.

Considering the arcs corresponding to the variables $Y_{ij} = 1$, we will get a connected subgraph spanning all the vertices of $\mathcal{N}$ because we have $n - 1$ arcs, one and only one arc incident in each vertex of $\mathcal{N} - \{1\}$. So, we have a feasible (integer) solution to (MLST). This spanning tree uses no more labels than the optimal value of (MLSTb) (or (MLSTc)), because we only put $Y_{ij} = 1$ for arcs such that X_{ij} is positive in the optimal solution of (MLSTb) (or (MLSTc)). So, variable v_l corresponding to the label of arc (i, j) is equal to one in the optimal solution of (MLSTb) (or (MLSTc)). Of course it cannot use fewer labels than the optimal value of (MLSTb) (or (MLSTc)), since the Y_{ij} obtained are feasible to (MLSTb) (or (MLSTc)).

3 Computational experiments

Several data sets were used to test the models that were introduced in the previous section. In all cases edge labels were represented as integer values uniformly generated between 1 and ℓ. First two sets of random networks:

1. of smaller instances with $n = 20, 50, 100$, $m = 4n$ and $\ell = 5, 10, 20$,
2. of larger instances with $n = 200, 500, 1000$, $m = 4n$ and $\ell = 5, 10, 20$,

were considered. Concerning these two data sets, the results presented in Tables 1 and 2 are average values obtained with 10 different instances generated for each dimension. Then grid networks with:

3. $n = 20, 36$, and with the forms 2×10, 4×5 and 2×18, 3×12, 6×6, respectively, and $\ell = 5, 10, 20$,

were considered. In all these instances at most 20 different labels were associated with the edges, having in mind possible practical applications in communication networks. However, a final data set where the numbers of vertices and of distinct labels coincides, following the specifications given in other works on heuristic methods for the MLSTP, was tested. Its characteristics were:

4. $n = 20, 50$, $m = d * n(n - 1)/2$, $d = 0.2, 0.5, 0.8$, and $\ell = n$.

Concerning this data set, the results presented in Table 4 are average values obtained with 10 different instances generated for each dimension.

We used CPLEX 11.0 to solve all these models in a PC Intel Core™2, 2.4 GHz with 2GB of RAM. The results obtained for the four different data sets are shown in Tables 1-4.

ℓ	5	10	20	5	10	20	5	10	20
n		20			50			100	
MLSTa	0.14	0.33	0.62	0.67	1.54	4.76	1.57	5.73	29.84
MLSTb	0.03	0.04	0.06	0.13	0.26	1.38	0.18	1.03	3.35
MLSTc	0.01	0.05	0.07	0.09	0.44	1.35	0.27	1.15	7.47
MLST	0.07	0.35	0.34	0.93	2.43	5.77	2.86	9.26	22.61

Table 1: CPU times (in seconds) in random networks of Set 1

ℓ	5	10	20	5	10	20	5	10	20
n		200			500			1000	
MLSTa	16.74	31.33	182.54	520.15	834.65	2079.80	3391.75	15085.38	– (*)
MLSTb	0.50	2.56	14.95	1.34	9.48	136.43	2.11	43.24	621.11
MLSTc	1.22	5.86	34.33	3.61	38.00	370.53	5.43	131.83	1994.24
MLST	18.62	48.40	223.56	550.84	1309.04	6650.91	2225.82	9822.25	– (*)

(*) CPU time exceeded 100000 seconds.

Table 2: CPU times (in seconds) in random networks of Set 2

The mean CPU time to obtain v(MLSTa) is usually smaller than the mean CPU time to obtain v(MLST), but the opposite happens with some instances of Table 1 and the largest instances of Table 2. The mean CPU time to obtain v(MLSTb) is much smaller than the mean CPU time to obtain v(MLSTa). In the larger instances is around 300 times smaller, but the difference increases with the size of the networks. The mean CPU time to obtain v(MLSTc) is also much smaller than the mean CPU time to obtain v(MLST). In average is around 75 times smaller. For instances with 1000 vertices and 10 different labels the mean CPU time to obtain v(MLSTb) was 43.24 seconds and to obtain v(MLSTa) was 15085.38. For instances with 1000 vertices and 5 different labels the mean CPU time to obtain v(MLSTb) was 2.11 seconds and to obtain v(MLSTa) was 3391.75. For instances with 1000 vertices and 10 different labels the mean CPU time to obtain v(MLSTc) was 131.83 seconds and to obtain v(MLST) was 9822.25. For instances with 1000 vertices and 5 different labels the mean CPU time to obtain v(MLSTc) was 5.43 seconds and to obtain v(MLST) was 2225.82.

ℓ	5	10	20	5	10	20	5	10	20	5	10	20	5	10	20
$p \times q$		2×10			4×5			2×18			3×12			6×6	
MLSTa	0.00	0.03	0.02	0.02	0.02	0.14	0.03	0.06	0.41	0.02	0.41	0.23	0.00	0.39	0.41
MLSTb	0.01	0.00	0.00	0.00	0.02	0.03	0.02	0.01	0.08	0.02	0.03	0.23	0.00	0.06	0.02
MLSTc	0.00	0.02	0.00	0.02	0.00	0.05	0.00	0.01	0.09	0.02	0.01	0.30	0.02	0.05	0.06
MLST	0.02	0.02	0.02	0.02	0.01	0.05	0.02	0.19	0.25	0.00	0.23	0.75	0.02	0.23	0.20

Table 3: CPU times (in seconds) in grid networks of Set 3

Usually it is faster to obtain v(MLSTb) than v(MLSTc), but the opposite happens in some cases, namely those in Set 4 as we can see in Table 4.

d	0.2	0.5	0.8	0.2	0.5	0.8
n		20			50	
MLSTa	0.08	0.93	1.53	63.22	96.34	73.48
MLSTb	0.02	0.13	0.37	10.31	8.60	7.77
MLSTc	0.03	0.08	0.24	6.43	9.21	4.38
MLST	0.06	0.48	0.97	24.07	104.71	102.07

Table 4: CPU times (in seconds) in random networks of Set 4

4 Conclusions

We focused on the minimum label spanning tree problem. This is an NP-hard problem, the goal of which consists of determining a spanning tree with the minimum number of different labels, assuming each edge to be associated with a label. We presented mixed integer formulations for the MLSTP and proved that one of their relaxation always gives the optimal value. The introduced formulations were tested on randomly generated instances of different types. For any of the formulations the CPU times demanded to find the optimal value depended on the size of the network, but specially on the number of distinct labels of network edges. Still, it was possible to solve the MLSTP in networks with 1000 vertices and 10 different labels in less than 44 seconds, and in networks with 50 vertices and 50 different labels in less than 8 seconds.

References

[1] R. Cerulli, A. Fink, Gentili M, and S. Voss. Metaheuristics comparison for the minimum labelling spanning tree problem. In B. L. Golden, S. Raghavan, and E. Wasil, editors, *The Next Wave on Computing, Optimization, and Decision Technologies*, volume 29, pages 93–106. Springer US, New York, 2005.

[2] R. Chang and S.-J. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.

[3] S. Consoli, J. A. Moreno, N. Mladenović, and K. Darby-Dowman. Constructive heuristics for the minimum labelling spanning tree problem: a preliminary comparison. Technical Report DEIOC-4, Universidad de La Laguna, La Laguna, September 2006. (http://hdl.handle.net/2438/504).

[4] S. Krumke and H. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.

[5] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50, 1956.

[6] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[7] Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84(2):99–101, 2002.

[8] Y. Xiong. *The minimum labeling spanning tree problem and some variants.* PhD thesis, Graduate School of the Univ. of Maryland, USA, 2005.

[9] Y. Xiong, B. Golden, and E. Wasil. Worst-case behavior of the MVCA heuristic for the minimum labeling spanning tree problem. *Operations Research Letters*, 33(1):77–80, 2005.

[10] Y. Xiong, B. Golden, E. Wasil, and S. Chen. The label-constrained minimum spanning tree problem. In S. Raghavan, B. Golden, and E. Wasil, editors, *Telecommunications Modeling, Policy, and Technology. Operations Research/Computer Science Interfaces*, volume 44, pages 39–58. Springer, United States, 2008.