On the Weight-Constrained Minimum Spanning Tree Problem

Agostinho Agra[1], Adelaide Cerveira[2], Cristina Requejo[1], and Eulália Santos[3]

[1] CIDMA and Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal
{aagra,crequejo}@ua.pt
[2] CIO and Department of Mathematics, University of Trás-os-Montes and Alto Douro, 5001-801 Vila Real, Portugal
cerveira@utad.pt
[3] CIDMA and School of Technology and Management, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal
eulalia.santos@ipleiria.pt

Abstract. We consider the weight-constrained minimum spanning tree problem which has important applications in telecommunication networks design. We discuss and compare several formulations. In order to strengthen these formulations, new classes of valid inequalities are introduced. They adapt the well-known cover, extended cover and lifted cover inequalities. They incorporate information from the two subsets: the set of spanning trees and the knapsack set. We report computational experiments where the best performance of a standard optimization package was obtained when using a formulation based on the well-known Miller-Tucker-Zemlin variables combined with separation of cut-set inequalities.

1 Introduction

Consider an undirected complete graph $G = (V,E)$, with node set $V = \{0,1,\ldots,n-1\}$ and edge set $E = \{\{i, j\},\ i, j \in V, i \neq j\}$. Associated with each edge $e = \{i, j\} \in E$ consider nonnegative integer costs c_e and nonnegative integer weights w_e. The Weight-constrained Minimum Spanning Tree problem (WMST) is to find a spanning tree $T = (V,E_T)$, $E_T \subseteq E$, in G of minimum cost $C(T) = \sum_{e\in E_T} c_e$ and with total weight $W(T) = \sum_{e\in E_T} w_e$ not exceeding a given limit H. This combinatorial optimization problem is weakly NP-hard [1].

The WMST is known under several different names. It was first mentioned in [1] as the *MST problem subject to a side constraint*. In this paper the authors propose an exact algorithm that uses a Lagrangian relaxation combined with a branch and bound strategy. A similar approach can also be found in [10]. Approximation algorithms were developed in [9,4,3]. In [3] the results in [9] are improved. A branch-and-bound algorithm for the weight-constrained maximum spanning tree problem was developed in [11].

The WMST appears in several real applications and the weight restrictions are mainly concerned with a limited budget on installation/upgrading costs. A classical application arises in the areas of communication networks and network design, in which information is broadcast over a minimum spanning tree. The upgrade and/or the design of the physical system is usually restricted to a pre-established budget.

J. Pahl, T. Reiners, and S. Voß (Eds.): INOC 2011, LNCS 6701, pp. 156–161, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In this paper we intend to fill a gap, i.e., the lack of research on formulations and valid inequalities for the WMST problem. Firstly we discuss extended formulations that are adapted from formulations for the Minimum Spanning Tree problem: the multicommodity flow formulation and formulations based on the well-known Miller-Tucker-Zemlin (MTZ) inequalities. These formulations are compared, from the computational point of view, with the classical cut-set formulation for the MST. Computational experiments show that interesting results can be obtained when a MTZ based reformulation [2] is combined with separation over the cut-set inequalities. Secondly, we discuss valid inequalities for the set of feasible solutions that take into account properties from the two subsets, the knapsack set and the set of spanning trees, simultaneously. These inequalities adapt for the WMST problem the well-known cover, extended cover and lifted cover inequalities.

In Section 2 we discuss formulations for the WMST problem while in Section 3 we discuss valid inequalities. In Section 4 we report some computational experiments.

2 Formulations

A natural way to formulate the WMST problem is to use a formulation for the Minimum Spanning Tree (MST) problem [6] and add the weight constraint $\sum_{(i,j)\in A} w_{ij}x_{ij} \leq H$.

It is well-known (see [6]) that oriented formulations (based on the underlying directed graph) for the MST lead, in general, to tighter formulations (formulations whose lower bounds provided by the linear relaxations are closer to the optimum values). Thus, in this section we consider the corresponding directed graph, with root node 0, where each edge $e = \{0, j\} \in E$ is replaced with arc $(0, j)$ and each edge $e = \{i, j\} \in E, i \neq 0$, is replaced with two arcs, (i, j) and (j, i), yielding the arc set $A = \{(i, j),\ i \in V \setminus \{0\}, j \in V, i \neq j\}$. These arcs inherit the cost and weight of the ancestor edge.

The two classical formulations on the space of the original variables (the binary variables x_{ij}, for all $(i, j) \in A$, indicating whether arc (i, j) is chosen or not) for the WMST, one using the circuit elimination inequalities and the other the cut-set inequalities to ensure connectivity/prevent circuits can be considered. The linear relaxation of both models provide the same bound [6]. We use the formulation with the cut-set inequalities, Cut-Set formulation, denoted by CS. As the number of cut-set inequalities increases exponentially with the size of the model, these inequalities are introduced in the model as cuts using separation. However, it is well-known that in order to ensure connectivity/prevent circuits, instead of using one of those families of inequalities with an exponential number of inequalities, one can use compact extended formulations. That is the case of the well-known Multicommodity Flow (MF) formulation where connectivity of the solution is ensured through the flow conservation constraints together with the connecting constraints [6] and the case of the well-known MTZ formulation where connectivity of the solution is ensured through the node position variables [2].

As stated above, all these four formulations can be used directly to formulate the WMST by adding the weight constraint. Next we propose one more extended formulation, based on the MTZ variables, requiring some additional elaboration. In addition to the binary variables x_{ij} defining the topology of the solution, we consider variables p_i, $i \in V$, which specify the weighted-position of node i in the tree, i.e. the sum of the weights of the arcs in the path between the root node and node i. The Weighted Miller-Tucker-Zemlin (WMTZ) formulation is as follows:

$$\begin{aligned}
min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} && \\
s.t. \quad & \sum_{i\in V} x_{ij} = 1 & j \in V \setminus \{0\} & \quad (1)\\
& \sum_{(i,j)\in A} w_{ij}x_{ij} \leq H & & \quad (2)\\
& w_{ij}x_{ij} + p_i \leq p_j + H(1-x_{ij}) & (i,j) \in A & \quad (3)\\
& 0 \leq p_i \leq H & i \in V & \quad (4)\\
& x_{ij} \in \{0,1\} & (i,j) \in A. & \quad (5)
\end{aligned}$$

Constraints (1) ensure that there is one arc incident to each node, with the exception of the root node. Constraint (2) is the weight constraint. Constraints (4) impose bounds on variables p_i. Constraints (3) prevent circuits and act as the well-known subtour elimination constraints given in [7] for the Traveling Salesman Problem: adding (3) for a circuit $\mathscr{C}$ ($x_{ij} = 1, (i,j) \in \mathscr{C}$) one obtains $\sum_{(i,j)\in\mathscr{C}} w_{ij} \leq 0$. On the other hand for any feasible weighted tree one can always find values for $p_j, \forall j \in N$, such that (3) and (4) are satisfied. Setting p_j to the weight of the path from the root node to any node j, ($p_j = p_i + w_{ij}$ for all (i,j) such that $x_{ij} = 1$ and $p_0 = 0$), then constraints (4) and (3), for all (i,j) such that $x_{ij} = 0$, are implied by the knapsack constraint (2).

Following Gouveia [2], constraints (3) can be lifted into several sets of inequalities. Computational results indicate that among all the lifted inequalities better computational results are obtained when the following inequalities

$$(H - w_{ji})x_{ji} + w_{ij}x_{ij} + p_i \leq p_j + H(1-x_{ij}) \qquad (i,j) \in A \qquad (6)$$

are incorporated in the formulation. Thus, henceforward we consider the WMTZ formulation with inequalities (3) replaced by (6).

3 Valid Inequalities

In order to strengthen the formulations presented in the previous section we discuss classes of valid inequalities.

We denote by X the set of feasible solutions to WMST. Set X can be regarded as the intersection of two well-known sets: $X = X_T \cap X_K$, where X_T is the set of spanning trees and X_K is the binary knapsack set defined by (2) and (5). Valid inequalities for X_T and valid inequalities for X_K are valid for X. While the polyhedral description of the convex hull of X_T, $P_T = conv(X_T)$ is well-known, see [6], for the polyhedral characterization of the convex hull of knapsack sets, $P_K = conv(X_K)$, only partial descriptions are known. This polyhedron is probably one of the most combinatorial optimization polyhedra studied. Kaparis and Letchford [5] present a very complete study on separation of valid inequalities for P_K. As in general $P = conv(X)$ is strictly included in $P_T \cap P_K$, there are fractional solutions that cannot be cut off by valid inequalities derived for P_T

or P_K. Hence, here we focus on valid inequalities derived for P that take into account properties from the two sets, simultaneously.

We call a set $C \subset E$ a Tree-Completion (TC) cover if for every spanning tree $T = (V, E_T)$ such that $C \subset E_T$, $W(T) = \sum_{e \in E_T} w_e > H$.

Proposition 1. *Given any TC cover C, the tree completion cover inequality (TCCI) $\sum_{e \in C} x_e \leq |C| - 1$ is valid for X.*

It can be checked that every cover inequality is a Subtour breaking Constraint (SC), or a TCCI or it is dominated by a SC or a TCCI. As for cover inequalities, TCCI are in general weak. One possible approach to strengthen these inequalities is by lifting. Given a TCCI, $\sum_{e \in C} x_e \leq |C| - 1$, a valid inequality $\sum_{e \in C} x_e + \sum_{e \in E \setminus C} \beta_e x_e \leq |C| - 1$, with $\beta_e \geq 0, e \in E \setminus C$, is called a lifted TCCI (LTCCI). One first approach to lift a TCCI is to adapt the well-known extended cover inequalities.

Proposition 2. *Let $C \subset E$ be a TC cover and let $C' = \{e \in E | w_e \geq \max\{w_f : f \in C\}$ and $C \cup \{e\}$ forms a cycle$\}$. The extended TCCI (ETCCI), $\sum_{e \in C} x_e + \sum_{e \in C'} x_e \leq |C| - 1$, is valid for X.*

LTCCIs can also be obtained via sequential lifting where the coefficients β_e are computed one at a time. Given a TCCI, $\sum_{e \in C} x_e \leq | C | -1$, and a LTCCI $\sum_{e \in C} x_e + \sum_{e \in R} \beta_e x_e \leq |C| - 1$, one can lift x_f, with $f \in E \setminus (C \cup R)$, by computing β_f, such that: $0 \leq \beta_f \leq f(C, R, \beta) = \min\{| C | -1 - \sum_{e \in C} x_e - \sum_{e \in R} \beta_e x_e : x \in X, x_f = 1\}$.

In order to derive LTCCIs no variables are fixed a priori (no restrictions are considered). However, a new class of lifted inequalities can be derived by the "usual" lifting procedure where the lifting is done by fixing the value of a set of variables, deriving a valid inequality for the restricted set that results from the variable fixing, and then sequentially lift each variable whose value has been fixed. We call the inequalities obtained by this procedure the Generalized Lifted TCCI Inequalities (GLI). To derive a GLI one fix a set E_0 of variables to zero, a set E_1 of variables to one, then generate a LTCCI for the restricted set ($X^R = X \cap \{x : x_e = 0, e \in E_0, x_e = 1, e \in E_1\}$) and lift sequentially all the variables with null coefficient in the LTCCI.

4 Computational Experiments

To compare the proposed formulations and test the valid inequalities, a test set of instances was generated. The costs and weights of a first test set with up to 100 nodes, were generated based on Pisinger's [8] spanner instances. In a second set of instances, with nodes from 150 to 300, the costs were based on Euclidean distances while weights were randomly generated in $[1, 100]$. In all instances H was fixed to the average between the minimum and the maximum weight spanning tree. All the tests were run on a Intel(R) Core(TM)2Duo CPU 2.00GHz with 1.99Gb of RAM and using the optimization software Xpress 7.1. To solve large size instances we focused on the comparison of the two hybrid procedures that result from the MTZ and WMTZ formulations

Table 1. Average gaps and average running times with formulations MTZ, WMTZ, MF and CS and the two hybrid procedures MTZ+C and WMTZ+C

$\lvert V \rvert$	LPgap MTZ	time MTZ	LPgap WMTZ	time WMTZ	LPgap MF	time MF	LPgap CS	time CS	LPgap MTZ+C	time MTZ+C	LPgap WMTZ+	time WMTZ+C
10	10.3	0.08	10.3	0.09	1.2	0.02	1.2	0.01	1.2	0.08	1.2	0.01
20	7.4	0.84	7.4	0.99	1.0	0.72	1.0	0.12	1.0	0.05	1.0	0.05
40	3.7	2236	3.7	2276	0.5	12.3	0.5	71.2	0.5	0.27	0.5	0.12
60	2.2	4335	2.2	4329	0.5	144.2	0.5	6269	0.5	0.6	0.5	0.36
80	1.7	4655	1.7	2868	1.0	306	1.0	*	1.0	1.46	1.0	1.25
100	1.3	4333	1.3	4326	0.3	3517	0.3	*	0.3	4329	0.3	2173
150									0.03	4.5	0.03	3.6
200									0.04	13.0	0.04	11.6
250									0.04	37.2	0.04	25.0
300									0.05	44.2	0.05	41.4

strengthened at the root node of the Branch and Bound with the cut-set inequalities, MTZ+C and WMTZ+C. Using the two hybrid procedures we solved all the generated instances up to 300 nodes. Table 1 gives the average integrality gap and the average running times (in seconds) for a set of 5 instances for each number of vertices. An asterisk means that some of the 5 instances were not solved within a time limit of one day.

For valid inequalities we tested separation heuristics for TCCIs, ETCCIs, LTCCIs and GLIs. Since the tested separation heuristics for GLIs provided no improvement when compared to the LTCCIs case, we do not discuss here the separation of GLIs. For separation of TCCIs we: (i) sort the edges accordingly to a given order; (ii) following that order, include one edge at a time into set C and check whether C defines a TC cover, and, if so, checks if the TCCI cuts off the fractional solution; (iii) if no cuts were found, and there are different orders to be tested, then return to (i) and use the next order to find cuts, otherwise STOP. For Step (i) we tested four (non-increasing) orderings of the edges based on the values: x_e^* (fractional solution); w_e; $(1-x_e^*)/w_e$ and $w_e x_e^*$. Although not reported here, tests using only one ordering showed that the best bounds were obtained using $w_e x_e^*$. Separation of ETCCIs was done similarly. When a cut from a TCCI is found we lift the TCCI into an ETCCI. For LTCCIs, when a cut from a TCCI is found we lift the variables accordingly to the order used to find the TCCI. The lifting coefficient of x_e is given by $\beta_e = \max\{0, \lceil \underline{f}(C,R,\beta) \rceil\}$ where $\underline{f}(C,R,\beta)$ denotes the value of the linear relaxation of $f(C,R,\beta)$.

For a selected set of 21 instances from 10 to 80 nodes we obtained an average integrality gap of 2.1%, 0.94%, 0.9% and 0.75% with the linear relaxation (LP), LP with TCCI cuts, LP with ETCCI cuts and LP with LTCCI cuts, respectively. To obtain the linear relaxation we used the MF formulation. For the linear relaxation of $f(C,R,\beta)$ we used the WMTZ formulation.

Acknowledgement

This work was partially supported by the Portuguese Foundation for Science and Technology, co-financed by the European Community Fund FEDER/POCI 2010.

References

1. Aggarwal, V., Aneja, Y.P., Nair, K.P.K.: Minimal spanning tree subject to a side constraint. Computers and Operations Research 9, 287–296 (1982)
2. Gouveia, L.: Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. Computers and Operations Research 22(9), 959–970 (1995)
3. Hassin, R., Levin, A.: An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. SIAM Journal on Computing 33(2), 261–268 (2004)
4. Hong, S.P., Chung, S.J., Park, B.H.: A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem. Operations Research Letters 32, 233–239 (2004)
5. Kaparis, K., Letchford, A.: Separation algorithms for 0-1 knapsack polytopes. Mathematical Programming 124(1–2), 69–91 (2010)
6. Magnanti, T., Wolsey, L.: Optimal trees. In: Ball, M., Magnanti, T., Monma, C., Nemhauser, G. (eds.) Network Models. Handbooks in Operations Research and Management Science, vol. 7, pp. 503–615. Elsevier Science Publishers, North-Holland, Amsterdam (1995)
7. Miller, C., Tucker, A., Zemlin, R.: Integer programming formulations and travelling salesman problems. Journal of the Association for Computing Machinery 7, 326–329 (1960)
8. Pisinger, D.: Where are the hard knapsack problems? Technical Report 2003/08, DIKU, University of Copenhagen, Denmark (2003)
9. Ravi, R., Goemans, M.: The constrained minimum spanning tree problem. In: Karlsson, R., Lingas, A. (eds.) SWAT 1996. LNCS, vol. 1097, pp. 66–75. Springer, Heidelberg (1996)
10. Shogan, A.: Constructing a minimal-cost spanning tree subject to resource constraints and flow requirements. Networks 13, 169–190 (1983)
11. Yamada, T., Watanabe, K., Kataoka, S.: Algorithms to solve the knapsack constrained maximum spanning tree problem. International Journal of Computer Mathematics 82, 23–34 (2005)