MINIMUM WEIGHT SPANNING TREES WITH BOUNDED DIAMETER

N.R. Achuthan and L. Caccetta

School of Mathematics and Statistics
Curtin University of Technology
GPO Box U1987
Perth, 6001
Western Australia.

ABSTRACT

Let G be a simple graph with non-negative edge weights. Determining a minimum weight spanning tree is a fundamental problem that arises in network design and as a subproblem in many combinatorial optimization problems such as vehicle routing. In some applications, it is necessary to restrict the diameter of the spanning tree and thus one is interested in the problem :

Find, in a given weighted graph G, a minimum weight spanning tree of diameter at most D.

This problem is known to be NP-complete for $D \geq 4$. In this paper we present a mixed integer linear programming formulation and discuss some solution procedures.

1. INTRODUCTION

Many combinatorial optimization problems involve determining a minimum weight restricted spanning tree as a subproblem. Papadimitriou and Yannakakis [7] discuss the complexity of such problems. In this paper we discuss the problem of determining a **Minimum Weight Spanning Tree with Bounded Diameter D**. We refer to this problem by **MWST-D**. The problem without the diameter restrictions can be solved very efficiently (see Gabow et al [4]).

For the most part, our graph theory notation and terminology follows that of Bondy and Murty [2]. Let G = (V,E) denote a finite and that G is connected and let $|V| = n$ and $|E| = m$. Associated with every edge (x,y) of E, there is a non-negative weight $w(x,y)$. The **distance** $d(x,y)$ between two vertices x and y in G is the number of edges in the shortest (x,y) - path in G. The **diameter** $d(G)$ of G is defined as the maximum distance in G; that is

$$d(G) = \max_{x,y \in V} \{d(x,y)\} .$$

The **eccentricity** $e(x)$ of a vertex x in G is defined as the distance of a vertex furtherest away from x; that is

$$e(x) = \max_{u \in V} \{d(x,u)\} .$$

For a given spanning tree T of G, we define the weight $w(T)$ of T by

$$w(T) = \sum_{(x,y) \in T} w(x,y) .$$

For convenience we denote the edge set of T also by T. Let

$$S_G = \{T : T \text{ is a spanning tree of } G\}.$$

Then we may express our problem as :

The MWST-D Problem: Given a simple undirected graph G with non-negative edge weights and a positive integer D, find a minimum weight spanning tree of diameter at most D. In otherwords, determine a spanning tree T^*, if it exists, such that

$$w(T^*) = \min \{w(T) : T \in S_G \text{ and } d(T) \le D\}.$$

Garey and Johnson [5] have shown that the MWST-D problem is NP-complete for any fixed $D \ge 4$. The problem can be easily solved in polynomial time for $D \le 3$. In this paper we provide a Mixed Integer Linear Programming (MILP) formulation of the MWST-D problem and discuss some solution procedures. The application of the MWST-D problem to network design is discussed in Caccetta [3].

2. MILP FORMULATION OF THE MWST-D PROBLEM

We begin by making the following two simple observations.

Observations :

(i) For any tree T in S_G,

$$d(T) \ge \max\{2, d(G)\}.$$

(ii) If $D < \max \{2, d(G)\}$, then the MWST-D problem is infeasible.

In our formulation we extend the given undirected graph G = (V,E) to a directed graph $G^* = (V^*, E^*)$ as follows. We add three new vertices s, t_1 and t_2 so that $V^* = V \cup \{s, t_1, t_2\}$. Every vertex x of V is joined to s, t_1 and t_2 by directed edges $(\vec{s,x})$, $(\vec{x,t_1})$ and $(\vec{x,t_2})$. For every in E^*. Call the resulting directed graph G^*. Observe that the indegree of s and the outdegrees of t_1 and t_2 are zero. We make G^* a weighted digraph by extending the weights of G as follows. If $(x,y) \in E$ then in G^* the directed edges (x,y) and (y,x) (we drop the arrow for convenience) have weights $w^*(x,y) = w^*(y,x) = w(x,y)$. For $x \in V$ we set

$$w^*(s,x) = 0$$

and

$$w^*(x,t_1) = w^*(x,t_2) = M,$$

where M is a large positive number; M is chosen such that it exceeds the value of the largest weight in G.

Associate a decision variable x_{ij} for every directed edge (i,j) in E^* and a real variable y_i for every vertex i in V^*. We define the integer L by

$$L = \begin{cases} \lfloor \frac{1}{2}D \rfloor + 2, & \text{if D is even} \\ \lfloor \frac{1}{2}D \rfloor + 3, & \text{otherwise.} \end{cases}$$

This definition of L helps in developing a spanning tree having a root node with minimum eccentricity. We first present the formulation and then discuss its justification.

Our MILP formulation is :

$$\text{Minimize } Z = \sum_{\substack{i \neq j \\ (i,j) \in E^*}} w(i,j)\, x_{ij} \qquad (1)$$

subject to the following constraints.

$$x_{ij} = 0 \text{ or } 1, \qquad \text{for } i \neq j \text{ and } (i,j) \in E^* \tag{2}$$

$$\sum_{i \in V} x_{si} = 1 \tag{3}$$

$$x_{si} + \sum_{\substack{k \in V \\ k \neq i}} x_{ki} = 1 \quad , \qquad \text{for all } i \in V \tag{4}$$

$$\sum_{\substack{k \in V \\ k \neq i}} x_{ik} + x_{it_1} + x_{it_2} \geq 1, \qquad \text{for all } i \in V \tag{5}$$

$$x_{si} + x_{it_1} + x_{it_2} \leq 1, \qquad \text{for all } i \in V \tag{6}$$

$$y_i - y_j + (L + 1)x_{ij} \leq L, \qquad \text{for all } i \neq j,\ (i,j) \in E^* \tag{7}$$

$$\sum_{i \in V} x_{it_1} \leq L - \lfloor \tfrac{1}{2} D \rfloor - 2 \tag{8}$$

$$y_{t_1} - y_s \leq L \sum_{i \in V} x_{it_1} \tag{9}$$

$$y_{t_2} - y_s \leq \lfloor \tfrac{1}{2} D \rfloor + 2 \tag{10}$$

Now to the justification of our formulation. We begin with the following simple facts.

Fact 1: Constraint (3) permits exactly one of the x_{sj}'s to be equal to 1.

Fact 2: Consider a solution of constraints (2) to (6). Restriction to the edges with $x_{ij} = 1$ gives rise to directed paths from s to t_1 or t_2 such that every vertex in these paths has indegree one (constraint (4)) and outdegree at least one (constraint (5)). Further, there may be some directed cycles not involving s and t. Figure 2.1 below illustrates the restriction of a solution of (2) to (6) to the edges of G^* with $x_{ij} = 1$ for the case when $x_{it_1} = 0$ for every $i \in V$.

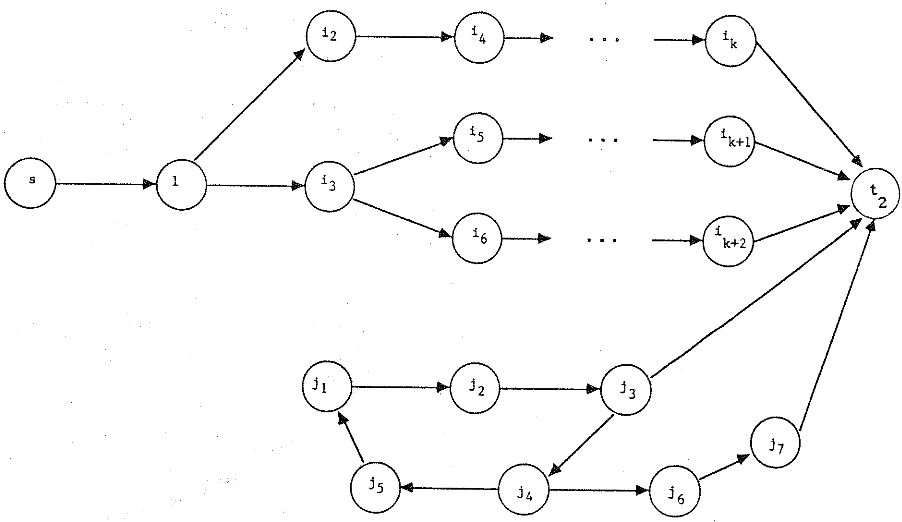

Figure 2.1 : Subgraph of G^* as a solution of (2) to (6)

Fact 3: In an optimal solution of the MILP (1) to (10) we have :

$$x_{ij} = 1 \Rightarrow x_{it_1} = x_{it_2} = 0 \quad \text{for } (i,j) \in E^*.$$

Fact 3 follows from the choice of $w^*(i,t_1) = w^*(i,t_2) = M$, a large positive number, and the objective of minimizing the function (1).

Having observed some simple facts, we next establish two important lemmas.

Lemma 2.1: Consider a solution of constraints (2) to (7). Restriction of this solution to the edges of G^* with $x_{ij} = 1$ gives rise to directed paths from s to t_1 or t_2 such that every vertex in these paths has indegree one and outdegree at least one. Further, there are no directed cycles.

Proof : In view of Fact 2 we need only establish that there is no directed cycle in the subgraph of G^* formed by taking the edges of G^* with $x_{ij} = 1$. Suppose $(j_1, j_2, \ldots, j_k, j_{k+1} = j_1)$ is a directed cycle such that

$$x_{j_r\, j_{r+1}} = 1, \text{ for } 1 \le r \le k.$$

From (7) we have

$$y_{j_1} - y_{j_2} + 1 \le 0$$

$$y_{j_2} - y_{j_3} + 1 \le 0$$

$$\vdots$$

$$y_{j_k} - y_{j_1} + 1 \le 0$$

eliminate directed cycles. □

Lemma 2.2: Suppose D is even. Consider a solution of constraints (2) to (10). Restriction of this solution to the edges of G^* with $x_{ij} = 1$ gives rise to directed paths from s to t_2 such that every path has at most (L-1) internal vertices.

Proof: Suppose $P = (i_0 = s, i_1, i_2, \ldots, i_K, i_{K+1} = t_2)$ is a directed path such that

$$x_{i_r i_{r+1}} = 1, \qquad \text{for } 1 \le r \le K.$$

Then from (7) we have

$$y_s - y_{i_1} + 1 \le 0$$

$$y_{i_1} - y_{i_2} + 1 \le 0$$

$$\vdots$$

$$y_{i_K} - y_{t_2} + 1 \le 0$$

Adding these yields

$$y_s - y_{t_2} + K + 1 \le 0.$$

Observe that since D is even constraint (8) forces $x_{it_1} = 0$ for every $i \in V$. Now using (10) we get

$$K + 1 \le y_{t_2} - y_s \le \lfloor \tfrac{1}{2} D \rfloor + 2 = L$$

and hence $K \le L - 1$. Thus the path P has at most $L - 1$ internal vertices. □

When D is odd we can, using the constraints (8) to (10) and the method of proof of Lemma 2.2, establish:

Lemma 2.3. Suppose D is odd. Consider a solution of constraints (2) to (10). Restriction of this solution to the edges of G* with $x_{ij} = 1$ gives rise to directed paths from s to t_1 or t_2 such that every path to $t_1(t_2)$ has at most $L - 1$ ($L - 2$) internal vertices. Moreover, there can be at most one directed path from s to t_1. □

Consider an optimal solution (x_{ij}, y_i) to the MILP problem (1) to (10). Define a directed graph $G' = (V^*, E')$ where

$$E' = \{(i,j) : (i,j) \in E^* \text{ and } x_{ij} = 1\}.$$

Figure 2.2 illustrates the structure of G′ for the case when $x_{it_1} = 0$ for all $i \in V$.

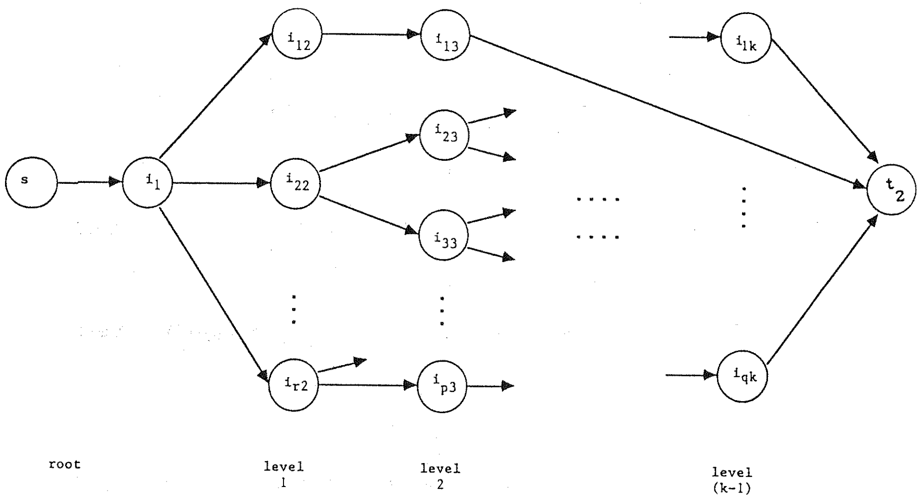

Figure 2.2: The Structure of G′

Lemmas 2.1, 2.2 and 2.3 together with Facts 1 to 3 yield the following theorem.

Theorem 2.1: The graph $G' = (V^*, E')$ defined above is a collection of directed paths from s to t_1 or t_2 such that :

(a) Vertex s has outdegree one;

(b) Every vertex of V has indegree one and outdegree at least one;

(c) Every directed path from s to t_1 or t_2 has at most (L-1) internal vertices.

(d) For D odd, every directed path from s to t_2 has at most (L-2) internal vertices. Moreover, there is at most one path from s to t_1. □

Consider an optimal solution (x_{ij}, y_i) to the MILP problem (1) to (10) and the graph G′ defined above. Denote by T′ the graph $G' - s - t_1 - t_2$. Clearly T′ is a spanning subgraph of G. In fact, it follows from the above discussion that if we ignore direction, then T′ is a spanning tree of G. We now prove that $d(T') \le D$.

Theorem 2.2: $d(T') \le D$.

Proof: In G′ let K denote the number of internal vertices in the longest directed path from s to t_1 or t_2. Clearly $d(T') \le 2(K-1)$. Since $2(K - 1) < D$ when $K < L-1$ we need only consider the case $K = L - 1$. For this case $2(K - 1) = D$ whenever D is even. When D is odd and $K = L - 1$ Theorem 2.1(c) and (d) implies that

$$d(T') \le (K - 1) + (K - 2) = 2L - 5 = D$$ □

Given any spanning tree T of G with $d(T) \le D$, we can construct a feasible solution (x_{ij}, y_i) to the MILP problem (1) to (10). The procedure is as follows.

Step 1: Set $x_{ij} = 0$ for every $(i,j) \in E$.

Step 2: Find the eccentricity e(j) for every $j \in V$.

Step 3: Find i^* such that $e(i^*) = \min_{j \in V} \{e(j)\}$.

Step 4: Set $x_{si*} = 1$, $x_{sj} = 0$ for all $j \neq i^*$, $y_s = 0$, $y_{i*} = 1$. Vertex i^* is said to be labelled.

Step 5: Choose a labelled vertex i (this means that y_i is fixed) and carry out the following steps :

(i) If $(i,j) \in T$ and j is not yet labelled, then set $x_{ij} = 1$ and $y_j = y_i + 1$. Vertex j is now labelled.

(ii) If there does not exist any j such that $(i,j) \in T$ and j is not labelled, then set $x_{it_2} = 1$.

Step 6: Repeat Step 5 until all vertices of G are labelled.

Step 7: Let $I = \left\{ i : y_i = \max_{j \in V} \{y_j\} \right\}$

If $|I| \geq 2$, then set $y_{t_1} = 0$ and $y_{t_2} = \max_{i \in V} \{y_i\} + 1$.

If $|I| = 1$, say $I = \{i^*\}$ then set $y_{t_1} = y_{i*}$,

$y_{t_2} = \max_{i \in V - i^*} \{y_i\}$, $x_{i*t_1} = 1$ and $x_{i*t_2} = 0$.

Now it is only a simple exercise to verify that the set $\{x_{ij}, y_i\}$ defined by the above procedure satisfy the conditions (2) to (10) and thus constitutes a feasible solution to the MILP problem (1) to (10). This together with the earlier results, establishes the following theorem.

Theorem 2.3 : The MILP formulation (1) to (10) solves the MWST-D problem. □

3. SOLUTION PROCEDURES

MILP problems have been the focus of considerable attention since the development of the Simplex Algorithm for solving linear programming problems. This is indeed evident from the vast literature that has accumulated over this time. It is not our intention to discuss the many algorithms that have been proposed to solve MILP problems. Rather, we give a brief discussion on how our MILP problem can be solved.

Exact solution procedures for MILP problems are usually extensions of the Branch and Bound methods for the Travelling Salesman problem (TSP); for a detailed discussion of such methods we refer to the book edited by Lawler et al [6]. In general, the branch and bound method for an optimization problem involves the decomposition of the given problem into a number of smaller sized subproblems. The important components of this procedure are branching, bounding and searching strategies.

The branching strategy dictates the manner in which a given problem is decomposed into two or more subproblems. The bounding strategy provides a bound on the objective function value with respect to a given subproblem. A level tree structure is constructed by representing every subproblem P by a node and linking a subproblem to its decomposed problems generated by applying the branching strategy to P. The search strategy determines how this tree is constructed by identifying the sequence of creating branches from the root node.

Further, it determines how much of the tree is actually constructed before an optimum solution is identified. We now discuss the branch and bound method for our problem.

Using our earlier notation, we may write our MWST-D problem as :

$$\min_{\substack{T \in S_G \\ d(T) \leq D}} \{w(T)\}$$

A node of the branch and bound tree structure represents a restricted problem based on the subgraph $G' = (V,E')$ of G (note $E' \subset E$). That is

$$\min_{\substack{T \in S_{G'} \\ d(T) \leq D}} \{w(T)\}$$

The root node, of course, corresponds to G itself; that is the original problem. A node of the tree can be recognized by G' or $S_{G'}$.

The lower bound associated with G' is determined as follows. First determine, using Kruskal's algorithm, the minimum weight spanning tree T' of G'. Obviously

$$w(T') \leq \min_{\substack{T \in S_{G'} \\ d(T) \leq D}} \{w(T)\} .$$

Thus $w(T')$ is a lower bound associated with the node corresponding to G'. If $d(T') \leq D$, then

$$w(T') = \min_{\substack{T \in S_{G'} \\ d(T) \leq D}} \{w(T)\} .$$

and so node G′ can be fathomed and the best known solution can be updated. If, on the other hand, d(T′) > D, then the node G′ will be maintained in the set of active nodes until either it is fathomed or its branches are created.

Among the set of all active nodes, the one with the least lower bound is chosen for further branching. Suppose we want to create branches of node G′ with d(T′) > D. Let u, v ∈ V be two vertices such that $d_{T'}(u,v) = d(T')$. Let $P = (u = u_0, u_1, \ldots, u_k = v)$ be the unique (u,v) - path in T′. We create k branches of node G′ as follows. Branch i will correspond to the graph $G'_i = G' - (u_{i-1}, u_i)$, $1 \le i \le k$.

The computational performance of the above procedure as well as a number of the methods including heuristic are currently under investigation. In fact, we are looking at a more general class of problems, the so called vehicle routing problem (see Achuthan and Caccetta [1]). We intend to report on the computational results in a subsequent paper.

ACKNOWLEDGEMENT

This work has been supported by ARC Grant A48932119.

REFERENCES

[1] N.R. Achuthan and L. Caccetta, Models for Vehicle Routing Problems, in Proceedings of the 10th National Conference of the Australian Society for Operations Research (G.P. Pervan, Editor) (1990), 276-294.

[2] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications. The MacMillan Press, London, 1977.

[3] L. Caccetta, Graph Theory in Network Design and Analysis, in Recent Studies in Graph Theory (V. Kulli, Editor) (1989), 29-63.

[4] H.N. Gabow, Z. Galil, T. Spencer and R.E. Tarjan, Efficient Algorithms for Finding Minimum Spanning Trees in Undirected and Directed Graphs, Combinatoria 2 (1986), 109 - 122.

[5] M.R. Garey and D.S. Johnson, Computers and Intractibility - A Guide to the Theory of NP-completeness. Freeman, San Francisco (1979).

[6] E. L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, The Travelling Salesman Problem. John Wiley & Sons, 1985.

[7] C.H. Papadimitriu and M. Yannakakis, The Complexity of Restricted Spanning Tree Problems, JACM 29 (1982), 285-309.