A Polyhedral Intersection Theorem for Capacitated Spanning Trees

by

Leslie A. Hall and Thomas L. Magnanti

OR 207-89 December 1989

A Polyhedral Intersection Theorem for Capacitated Spanning Trees

Leslie A. Hall
Dept. Civil Eng. and Operations Research
Princeton University

Thomas L. Magnanti
Sloan School of Management
M.I.T.

December 1989

Abstract

In a two-capacitated spanning tree of a complete graph with a distinguished root vertex v, every component of the induced subgraph on $V\backslash\{v\}$ has at most two vertices. We give a complete, non-redundant characterization of the polytope defined by the convex hull of the incidence vectors of two-capacitated spanning trees. This polytope is the intersection of the spanning tree polytope on the given graph and the matching polytope on the subgraph induced by removing the root node and its incident edges. This result is one of very few known cases in which the intersection of two integer polyhedra yields another integer polyhedron. We also give a complete polyhedral characterization of a related polytope, the 2-capacitated forest polytope.

1 Introduction

The κ-Capacitated Minimum Spanning Tree Problem is a capacitated verstion of the classical minimal spanning tree problem. Given a complete undirected graph $G = (V,\ E)$ defined on a vertex set $V = \{1, \ldots, n\}$, as well as a distinguished root vertex $1 \in V$, and costs on the edges $c : E \rightarrow \mathbb{R}$, find a minimum-cost spanning tree subject to the additional constraint that no subtree off of the root contains more than κ vertices (Figure 1).

For any value of $\kappa \geq n-1$, this problem is equivalent to the (regular) minimum spanning tree problem, for which efficient algorithms exist. However, for values of κ as small as 3 and as large as $n/2$, the κ-capacitated spanning tree problem is NP-hard in the strong sense [2, 10]; consequently, no polynomial algorithm exists for these cases unless $P = NP$ [8].

As is well known, the 2-capacitated vehicle-routing problem is equivalent to finding a matching in a related graph. Similarly, the 2-capacitated spanning tree problem is equivalent to a non-bipartite matching problem on a graph of comparable size, and is thus polynomially solvable. If n is odd, then finding a 2-capacitated tree in $G = (V,\ E)$ is equivalent to finding a minimum-cost perfect matching in a complete graph $\overline{G} = (\overline{V},\ \overline{E})$, defined on the vertex set $\overline{V} = \{2, \ldots, n\}$, and with edge costs

$$\hat{c}_{uv} = \min\{c_{1u} + c_{uv},\ c_{1v} + c_{uv},\ c_{1u} + c_{1v}\}.$$

If n is even, the addition of a dummy node $n+1$ and edge costs

$$\hat{c}_{v,n+1} = c_{1v}, \quad v = 2, \ldots, n$$

ensures the same equivalence between the 2-capacitated minimum spanning tree problem and a minimum-cost perfect matching problem. To see this equivalence for the case when n is odd, notice that we can always translate perfect matching in the new graph into a 2-capacitated tree for the original graph G in the following manner. Suppose that edge uv is in the matching. Then if $\hat{c}_{uv} = c_{1u} + c_{uv}$ (or $c_{1v} + c_{uv}$), we include edges $1u$ (or $1v$) and uv in the 2-capacitated tree; and if $\hat{c}_{uv} = c_{1u} + c_{1v}$, then we include edges $1u$ and $1v$ in the 2-capacitated tree. This transformation gives a

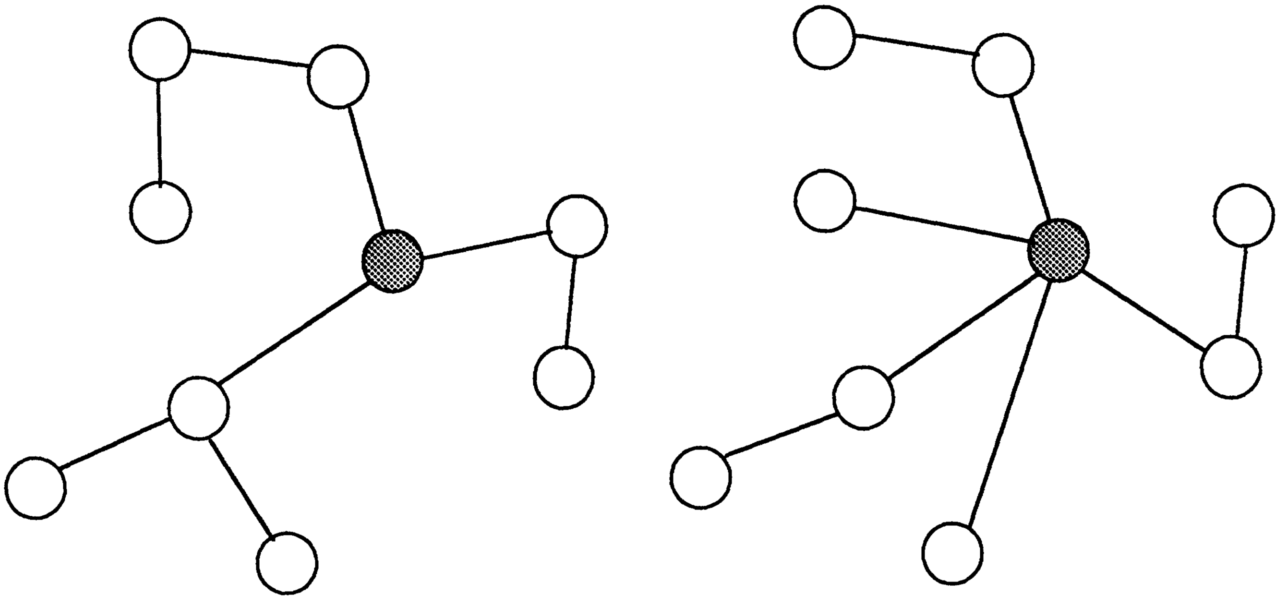

Figure 1: A 3-capacitated tree (left), and a 2-capacitated tree on the same graph. The shaded vertex is the root.

2-capacitated tree of cost equal to that of the given matching. Conversely, we can translate an optimal 2-capacitated tree into a perfect matching of equal cost for the new graph. Because each 2-capacitated tree contains an even number of single-vertex subtrees off of the root vertex, we can pair these vertices arbitrarily. Now we apply the inverse of the prior translation to these pairs and the 2-vertex components to obtain a perfect matching of equal cost. (Notice that, if the 2-capacitated tree is optimal, the edge costs will translate properly.) The equivalence for the case when n is even is similar.

Because the 2-capacitated spanning tree problem is polynomially solvable, we might expect that we can find an explicit linear-programming characterization of the problem; indeed, this is the case. In this paper, we present a complete characterization of the polytope defined by the incidence vectors of 2-capacitated spanning trees. What makes this characterization particularly satisfying is that it is essentially the intersection of two polyhedra representing graph structures closely related to this one: trees and matchings. In his pioneering work of the 1960's, Edmonds described both of these polyhedra [3, 4].

We also consider a combinatorial structure closely related to the 2-capacitated spanning tree problem—the 2-capacitated forest problem. A κ-capacitated forest,

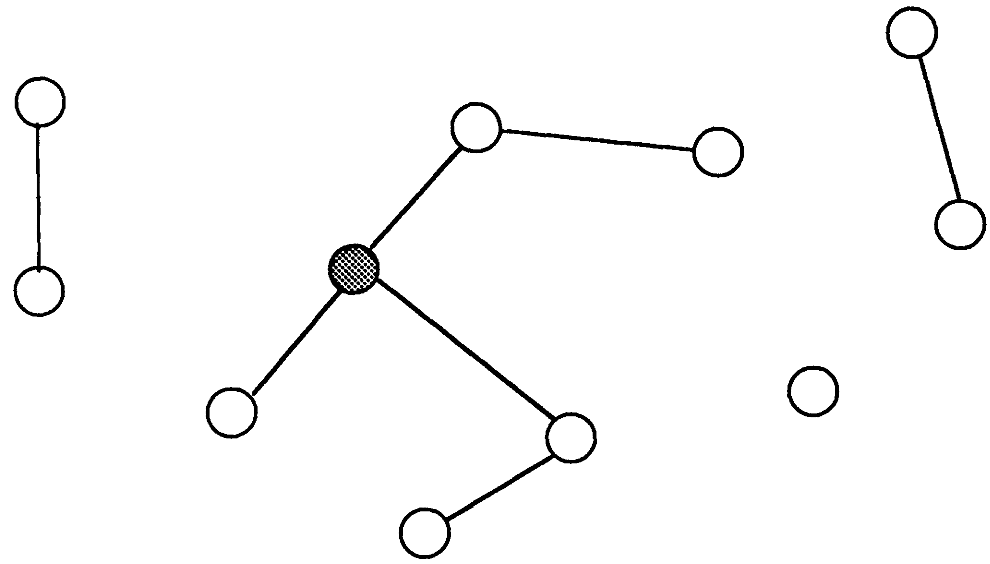

Figure 2: A 2-capacitated forest.

defined with respect to a root vertex, is a forest in which the connected component containing the root vertex is a κ-capacitated tree, and every other component contains at most κ vertices (Figure 2). In other words, a κ-capacitated forest is a κ-capacitated tree with some of the root edges (edges incident to the root vertex) removed. We present a complete characterization of this problem's associated polytope as well; it is closely related to the previous characterization.

In general, the intersection of two integral polyhedra does not have integral extreme points. A notable exception to this principle is the intersection of two polymatroids [5]. The forest-cover polytope provides another example [6]. A *forest cover* of a graph is a set of edges that form both a forest and a cover (a set of edges with the property that every vertex is incident to at least one edge) of the graph, *i.e.*, a forest containing no isolated vertices. Gamble and Pulleyblank [6] show that the forest-cover polytope is the intersection of the forest polytope and the cover polytope, both of which are easy to characterize polyhedrally.

In closing this section, we remark that the polyhedral structure of the κ-capacitated spanning tree problem for $\kappa > 2$ is quite complicated, not surprisingly. For a partial description of the associated polyhedron, as well as that for related vehicle routing problems, see Araque, Hall and Magnanti [1].

2 The Two-Capacitated Spanning Tree Polytope

We use the following notation throughout this paper. Let uv represent the undirected edge $\{u,v\} \in E$. For a subset of nodes $S \subseteq V$, let $E(S) = \{uv : u, v \in S\}$ denote the set of all edges between nodes of S, and for any vertex $v \in V$, let $\delta(v) = \{e = uv : u \in V\}$ denote the set of edges incident to v. Finally, let x be a vector in the space of edges of G, $x \in \mathbb{R}^{|E|}$. For any subset of edges $A \subseteq E$, we express the sum of the weights of the edges in A as $x(A) = \sum_{e \in A} x_e$.

The main result of this paper is the following theorem.

Theorem 1 The following set of inequalities and one equality is a complete description of the 2-capacitated spanning tree polytope.

$$\begin{aligned} x(E) &= n-1 && && (1)\\ x(\delta(v)\backslash\{1v\}) &\leq 1, && v \in V,\ v \neq 1 && (2)\\ x(E(S)) &\leq \left\lfloor \tfrac{|S|}{2} \right\rfloor, && S \subseteq V,\ |S| \geq 3,\ |S| \text{ odd},\ 1 \notin S && (3)\\ x(E(T)) &\leq |T|-1, && T \subseteq V,\ 2 \leq |T| \leq n-2,\ 1 \in T && (4)\\ x &\geq 0. && && (5) \end{aligned}$$

That is, the set of vertices of the polytope described by this system is exactly the set of incidence vectors of feasible 2-capacitated spanning trees.

Before proving this theorem, we observe that this formulation is actually the intersection of two well-studied polyhedra—the spanning tree polytope and the matching polytope. The following set of inequalities defines the polytope whose extreme points are the incidence vectors of trees in G [4].

$$\begin{aligned} x(E) &= n-1 && && (1)\\ x(E(T)) &\leq |T|-1, && T \subseteq V,\ |T| \geq 2 && (6)\\ x &\geq 0. && && (5) \end{aligned}$$

The following inequalities describe the polytope in $\mathbb{R}^{|E|}$ whose extreme points correspond to the incidence vectors of matchings in the subgraph $\overline{G}$ of G induced by removing the root vertex 1 and its incident edges, *i.e.*, $\overline{G} = (\overline{V}, \overline{E})$, $\overline{V} = \{2, \ldots, n\}$, $\overline{E} = E(\overline{V})$ [3].

$$\overline{x}(\delta(v)) \le 1, \quad v \in \overline{V} \tag{7}$$

$$\overline{x}(E(S)) \le \left\lfloor \frac{|S|}{2} \right\rfloor, \ S \subseteq \overline{V}, \ |S| \ge 3, \ |S| \text{ odd} \tag{8}$$

$$\overline{x} \ge 0. \tag{9}$$

If we consider the vector $\overline{x}$ as an element of $\mathbb{R}^{|E|}$ lying in the subspace corresponding to the set $\overline{E}$, then inequalities (7), (8), and (9) are equivalent to (2), (3), and (5), respectively. Let Q be the polytope described by (1) through (5). Then Q is the intersection of the tree polytope, given by (1), (6), and (5), and the polytope of matchings on the subgraph $\overline{G}$, given by (2), (3), and (5). To see this, we note that constraints (1), (2), (3), and (5) appear explicitly in the description of Q; and each inequality of (6) either is a member of (4) or is dominated by an inequality of (3).

3 Proof of Theorem 1

Let P be the convex hull of incidence vectors corresponding to 2-capacitated trees, and let Q be the polyhedron given by (1) through (5). It is easy to see that (1) through (5) are valid constraints for P. Since extreme points of P represent trees, the elements of P satisfy constraints (1), (4), and (5). Moreover, in a 2-capacitated tree, no subtree off of the root vertex may contain more than 2 nodes; in particular, if we delete all edges incident to the root vertex 1, the resulting components contain at most 2 nodes. Thus these remaining edges correspond to a matching in the induced subgraph $\overline{G}$, and so constraints (2) and (3) are also valid for extreme points of P. Thus $P \subseteq Q$.

It remains to show that $Q \subseteq P$, *i.e.*, that if $x \in Q$ then x can be expressed as a convex combination of 2-capacitated trees. Let $\overline{P}$ be the matching polyhedron on $\overline{G}$. If $x \in Q$, then the partial vector $\overline{x}$ given by projecting x into the subspace $\mathbb{R}^{|\overline{E}|} \subseteq \mathbb{R}^{|E|}$, *i.e.*,

$$\overline{x} = (x_e \ : \ e \in \overline{E}),$$

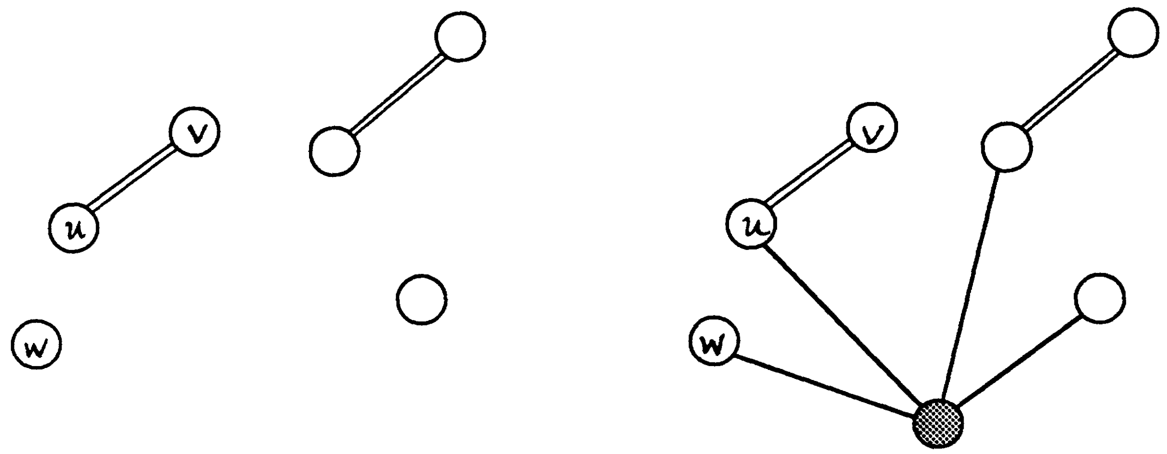

Figure 3: The (non-perfect) matching shown is extended to a 2-capacitated tree by adding root edges.

is contained in $\overline{P}$. This fact is easy to see by noting that $\overline{x}$ satisfies (7), (8), and (9), which describe $\overline{P}$ completely. Thus if $m^1, \ldots, m^L \in \mathbb{R}^{|\overline{E}|}$ are the 0-1 incidence vectors of matchings in $\overline{G}$, we can express $\overline{x}$ as a convex combination of matchings on $\overline{E}$,

$$\overline{x} = \sum_{i=1}^{L} \lambda_i m^i, \qquad \sum_{i=1}^{L} \lambda_i = 1, \quad \lambda \geq 0.$$

We will show that it is possible to use this convex combination of matchings for $\overline{x}$ to construct a representation of $x \in Q$ as a convex combination of 2-capacitated trees $\tau^1, \tau^2, \ldots, \tau^K$,

$$x = \sum_{i=1}^{K} \mu_i \tau^i, \qquad \sum_{i=1}^{K} \mu_i = 1, \quad \mu \geq 0.$$

We might view this task as follows. Consider any of the individual matchings m^i such as the one shown in Figure 3. We can transform this matching into a 2-capacitated tree τ^i by adding edge $1w$ if node w is unmatched and by adding either edge $1u$ or edge $1v$ if edge uv is in the matching. We can view this transformation as an allocation process. If the given matching m^i has weight λ_i, then the tree τ^i and thus the edges $1w$ and $1u$ (or $1v$) have a weight λ_i; we will associate, or allocate, a weight of λ_i from edge $1u$ (or edge $1v$) to edge uv, and allocate a weight of λ_i from edge $1w$ to node w. We need to make the allocation so that the sum of all allocations from any root edge $1v$ equals its weight x_{1v} in the given vector x. By doing so, we will have obtained the desired representation $x = \sum_{i=1}^{K} \mu_i \tau^i$.

Rather than making this allocation for each matching individually, we will make it in aggregate. For all $v \in \overline{V}$, let $M(v) \subseteq \{m^1, \ldots, m^L\}$, be those matchings in which vertex v is unmatched, and for all $e \in \overline{E}$, let $M(e)$ be the matchings containing edge e. Then for any edge $e \in \overline{E}$, e is contained in a fraction

$$\sum_{i \in M(e)} \lambda_i = x_e$$

of the weight of the matchings. For any vertex $u \in \overline{V}$, u is unmatched in a fraction

$$\sum_{i \in M(u)} \lambda_i = 1 - \sum_{v \in \overline{V}} x_{uv}$$

of the weight of the matchings. We would like to allocate the weight of the root edges to the matchings, in order to extend the matchings to 2-capacitated trees.

Lemma 1 For any $x \in Q$, there exists an allocation

$$\alpha : (E \setminus \overline{E} \times \overline{E} \cup \overline{V}) \longrightarrow [0,\ 1]$$

of weights from root edges $1v$ to edges $uv \in \overline{E}$ and nodes $v \in \overline{V}$ satisfying the following conditions:

$$\sum_{w \in \overline{V}} \alpha(1v,\ vw) + \alpha(1v,\ v) = x_{1v}, \qquad \forall\, 1v \in E \setminus \overline{E}; \tag{10}$$

$$\alpha(1v,\ v) = 1 - \sum_{u \in \overline{V}} x_{uv}, \quad \forall\, v \in \overline{V}; \tag{11}$$

$$\alpha(1v,\ uv) + \alpha(1u,\ uv) = x_{uv}, \qquad \forall\, uv \in \overline{E}. \tag{12}$$

If such an allocation α exists, we can use it to construct the desired convex combination of 2-capacitated trees for x as follows. To simplify the construction, we assume that all λ_i and all allocations $\alpha(1v, v)$ and $\alpha(1v, uv)$ are rational. (This assumption is not necessary; it simply allows us temporarily to clear fractions.) Let K be chosen so that $K\lambda_i \in \mathbf{Z}$, $i = 1, \ldots, L$, and $K\alpha(1v, v)$, $K\alpha(1v, uv) \in \mathbf{Z}$, for all

$u, v \in \overline{V}$. Consider a collection of matchings $\mathcal{C}$ consisting of $K\lambda_i$ copies of matching m^i, $i = 1, \ldots, L$. We interpret these matchings as incidence vectors of matchings $\phi \in \mathbb{R}^{|E|}$ in the original graph G. These matchings will be transformed progressively to forests, and finally to trees, by the following procedure. For each node $u \in \overline{V}$,

(1) For each forest $\phi \in \mathcal{C}$, if u is unmatched in ϕ then add edge $1u$ to ϕ (*i.e.*, change ϕ_{1u} from 0 to 1);

(2) For each node $v \in \overline{V}$, $v \neq u$, if node v has not yet been processed, then add edge $1u$ to $K\alpha(1u, uv)$ forests $\phi \in \mathcal{C}$ that contain edge uv, and add edge $1v$ to the remaining $K\alpha(1v, uv)$ forests $\phi \in \mathcal{C}$ that contain edge uv (*i.e.*, for every forest ϕ containing edge uv, either change ϕ_{1u} or ϕ_{1v} from 0 to 1, but not both).

Step 2 is possible because edge uv is contained in exactly

$$Kx_{uv} = K\alpha(1u, uv) + K\alpha(1v, uv) \tag{13}$$

forests of $\mathcal{C}$. Furthermore, when it teminates, the procedure has transformed each matching of $\mathcal{C}$ into a tree: for each original matching, Step 1 has attached every isolated node to the root, and by the observation (13), Step 2 has attached every 2-node component to the root.

Let us denote the tree completions of $\mathcal{C}$ by $\tau^1, \ldots, \tau^K$. We claim that

$$x = \sum_{i=1}^{K} (1/K)\tau^i. \tag{14}$$

For $e \in \overline{E}$, clearly $x_e = \sum_{i=1}^{K}(1/K){\tau^i}_e$, by the definition of the original matchings of $\mathcal{C}$. For any $u \in \overline{V}$, Steps 1 and 2 imply that $\sum_{i=1}^{K} {\tau^i}_{1u} = K\alpha(1u, u) + K\sum_{v\in\overline{V}} \alpha(1u, uv)$, and thus

$$\begin{aligned} x_{1u} &= \sum_{v\in\overline{V}} \alpha(1u, uv) \ + \alpha(1u, u) \\ &= (1/K)\sum_{i=1}^{K} {\tau^i}_{1u}, \end{aligned}$$

showing that (14) holds for the vector component corresponding to edge $1u$; thus (14) holds. Therefore, to prove Theorem 1 it is sufficient to prove Lemma 1.

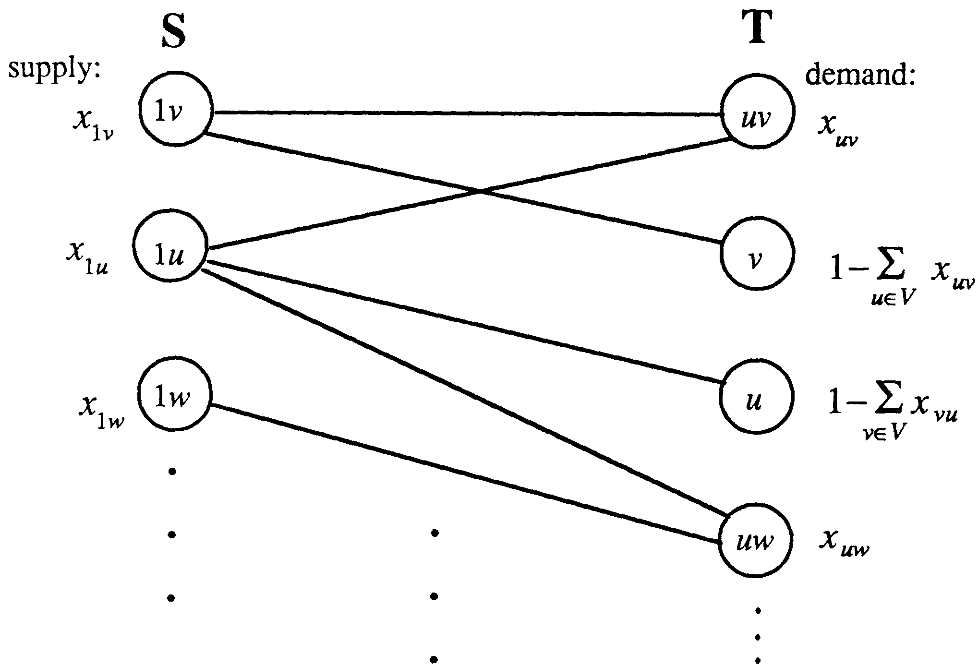

Figure 4: A feasible solution to this bipartite minimum-cost flow problem solves the α-allocation problem. S is the set of root-edges of G; T is the set of all non-root edges of G and all non-root vertices of G, with demands based on the values of x that need to be allocated to 2-capacitated trees.

Proof of Lemma 1

Finding the desired allocation is equivalent to finding a feasible solution to the bipartite flow problem shown in Figure 4: the "supplies" s_v for the set S of supply nodes correspond to the root-edge weights x_{1v}, and the "demands" d_e or d_v for the set T of demand nodes correspond to weights of edges $e \in \overline{E}$ and to one minus the sum of the weights of edges incident to node v, respectively. First, we note that

$$\begin{aligned}\sum_{v\in\overline{V}} s_v = \sum_{v\in\overline{V}} x_{1v} &= (n-1) - x(\overline{E}) \\ &= \sum_{v\in\overline{V}}(1 - \sum_{u\in\overline{V}} x_{vu}) + x(\overline{E}) \\ &= \sum_{v\in\overline{V}} d_v + \sum_{uv\in\overline{E}} d_{uv},\end{aligned}$$

i.e., total supply equals total demand. The bipartite flow problem is feasible if and only if the maximum flow problem in Figure 5 has value $x(E \setminus \overline{E}) = n - 1 - x(\overline{E})$.

Equivalently, we can show that the minimum cut has capacity $x(E \setminus \overline{E})$. Let c^* denote the capacity of the minimum cut. Evaluating the cut around the sink shows

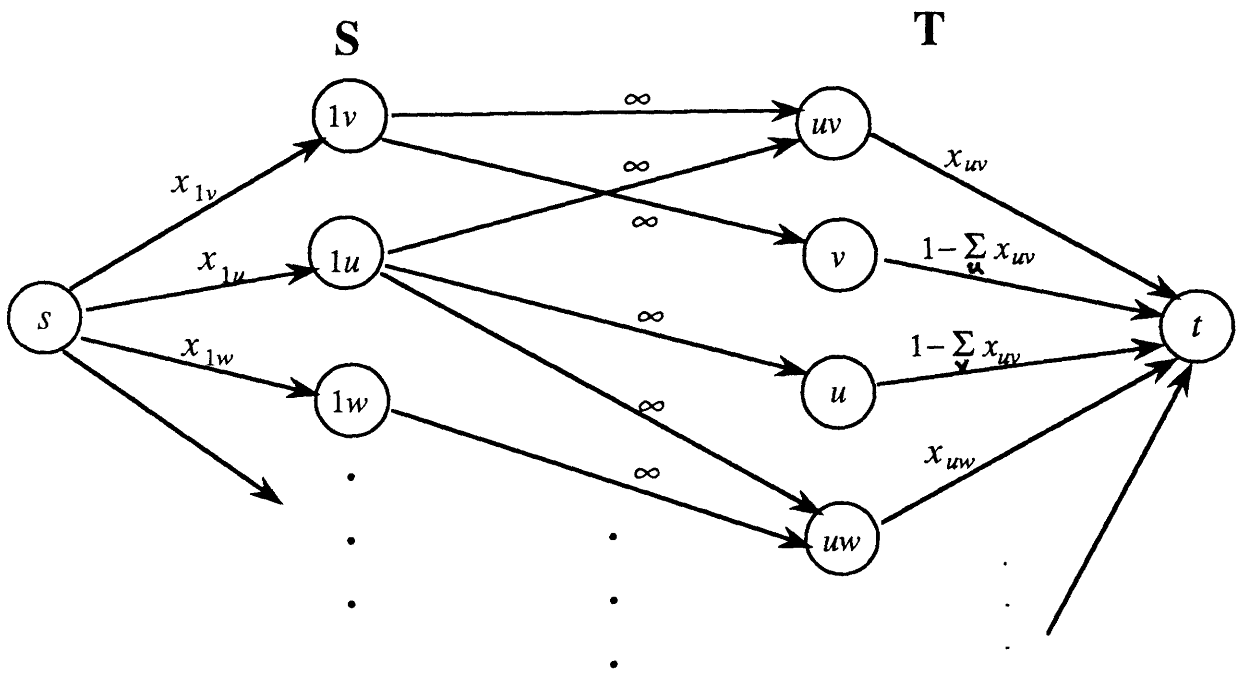

Figure 5: A maximum flow equal to $x(E)$ through this graph is equivalent to the minimum cost flow problem being feasible.

that

$$c^* \leq x(E \setminus \overline{E}) = \sum_{v \in \overline{V}} d_v + \sum_{uv \in \overline{E}} d_{uv}.$$

Thus, we need to show that no cut c has capacity $c < \sum_{v \in \overline{V}} d_v + \sum_{uv \in \overline{E}} d_{uv}$.

By a slight abuse of notation, we equate the vertices of the auxiliary bipartite graph in Figure 4 with the edges and vertices of the original graph G. Consider an arbitrary cut in the graph $\{\{s\} \cup (S \setminus Y) \cup (T \setminus Z)\ ;\ \{t\} \cup Y \cup Z\}$ (see Figure 6). We may assume without loss of generality that

$$\begin{aligned} uv \in Z &\implies 1v \in Y,\ 1u \in Y; \\ v \in Z &\implies 1v \in Y, \end{aligned} \qquad (A1)$$

since otherwise the capacity of the cut is infinite. In addition, we can make the following assumptions, which are not restrictive:

$$\begin{aligned} 1u \in Y &\implies u \in Z; \\ 1u, 1v \in Y &\implies uv \in Z. \end{aligned} \qquad (A2)$$

Conditions (A2) are not restrictive because adding these additional vertices to Z can only decrease, not increase, the capacity of the cut. For notational purposes, we

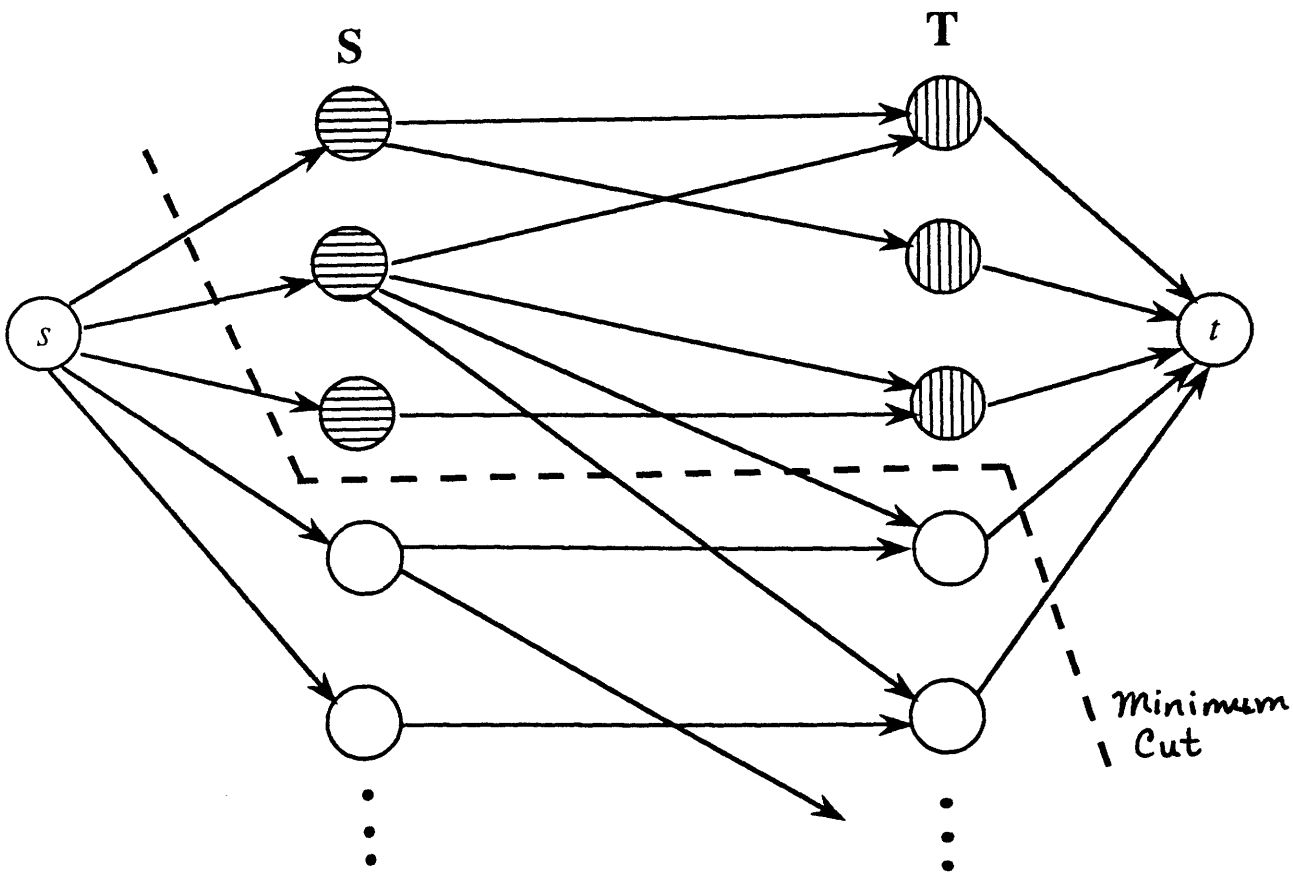

Figure 6: A diagram of the minimum cut in the bipartite graph. The horizontally striped vertices are the set Y, the vertically striped vertices the set Z. Note that the cut can contain no infinite-capacity arcs.

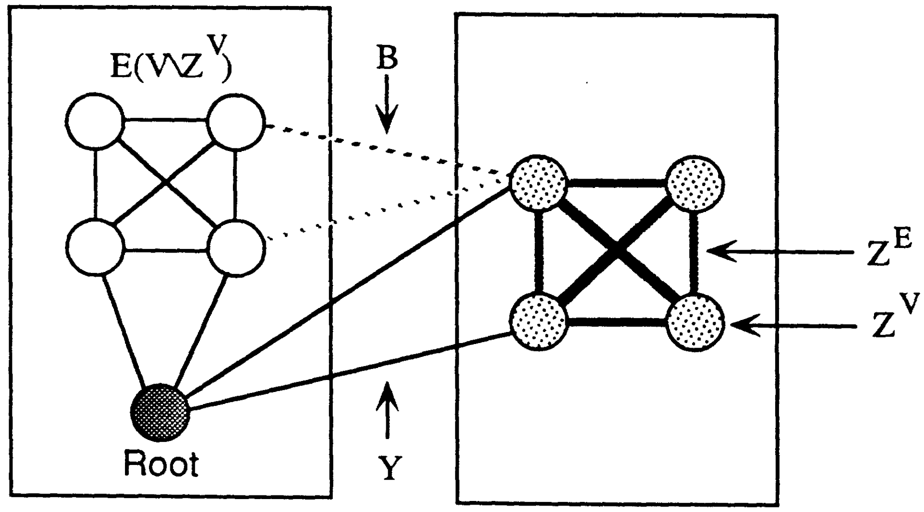

Figure 7: Network decomposition corresponding to a minimum cut.

partition Z as $Z = Z^V \cup Z^E$, according to whether an element of Z represents a vertex or an edge, respectively, in the original graph G; that is,

$$Z^V = \{v \in \overline{V} : v \in Z\}, \text{ and } Z^E = \{e \in \overline{E} : e \in Z\}.$$

Let B be the subset of edges from $\overline{E}$ containing exactly one endpoint in Z^V,

$$B := \{e = uv : u \in Z^V,\ v \notin Z^V\} \cap \overline{E}$$

(see Figure 7). A simple counting argument, coupled with our assumptions (A1) and (A2), shows that

$$\sum_{v \in Z^V} \sum_{u \in \overline{V}} x_{uv} = x(B) + 2x(Z^E) = x(B) + 2 \sum_{e \in Z^E} d_e.$$

Interpreted differently,

$$\begin{aligned} \sum_{v \in Z^V} \sum_{u \in \overline{V}} x_{uv} &= |Z^V| - |Z^V| + \sum_{v \in Z^V} \sum_{u \in \overline{V}} x_{uv} \\ &= |Z^V| - \sum_{v \in Z^V} (1 - \sum_{u \in \overline{V}} x_{uv}) \\ &= |Z^V| - \sum_{v \in Z^V} d_v. \end{aligned}$$

Combining these two equations gives

$$x(B) = |Z^V| - \sum_{v \in Z^V} d_v - 2 \sum_{e \in Z^E} d_e. \tag{15}$$

Next, we observe that the tree constraint (4) applied to the vertex set $V \setminus Z^V$ yields

$$x(E(V \setminus Z^V)) \le |V \setminus Z^V| - 1 = n - 1 - |Z^V|. \tag{16}$$

Recalling equation (1) and the fact that $E(V \setminus Z^V) \cup B \cup Z^E \cup Y = E$ (see Figure 7), we have

$$x(E(V \setminus Z^V)) = (n-1) - \left(x(B) + x(Z^E) + x(Y)\right). \tag{17}$$

Combining (17) and (16) and rearranging terms yields

$$x(Y) \ge |Z_V| - x(B) - x(Z^E).$$

Finally, substituting for $x(B)$ using (15) and for $x(Z^E)$ using $\sum_{e \in Z^E} d_e$, we have

$$\begin{aligned} \sum_{v \in Y} s_v = x(Y) &\ge |Z^V| - \left(|Z^V| - \sum_{v \in Z^V} d_v - 2 \sum_{e \in Z^E} d_e\right) - \sum_{e \in Z^E} d_e \\ &= \sum_{v \in Z^V} d_v + \sum_{e \in Z^E} d_e. \end{aligned}$$

Thus the capacity of the cut $\{\{s\} \cup (S \setminus Y) \cup (T \setminus Z)\ ;\ \{t\} \cup Y \cup Z\}$ in the max-flow problem (Figure 6) is

$$\begin{aligned} \sum_{v \in Y} s_v + \sum_{v \in \overline{V} \setminus Z^V} d_v + \sum_{e \in \overline{E} \setminus Z^E} d_e &\ge \sum_{v \in Z^V} d_v + \sum_{e \in Z^E} d_e + \sum_{v \in \overline{V} \setminus Z^V} d_v + \sum_{e \in \overline{E} \setminus Z^E} d_e \\ &= \sum_{v \in \overline{V}} d_v + \sum_{e \in \overline{E}} d_e. \end{aligned}$$

The capacity of this cut is indeed greater than or equal to $c^* = \sum_{v \in \overline{V}} d_v + \sum_{e \in \overline{E}} d_e$, as we wished to show, thus concluding the proof of Lemma 1.

Our previous remarks now imply that, indeed, $Q \subseteq P$, and so we have established Theorem 1. Q.E.D.

We also note that for $N \ge 4$ constraints (1) through (5) provide a non-redundant, as well as complete, characterization of P. This fact is easy to prove by observing that for each inequality of (2) through (5), some vector not in P satisfies every constraint of (1) through (5) except for the given constraint, and thus no constraint of (2) through (5) is redundant.

4 The Two-Capacitated Forest Polytope

The polytope Q is a face of the polytope that is given by replacing equation (1) with the inequality

$$x(E) \leq n-1 \tag{18}$$

in the polyhedral description of Q. This new polytope, given by (2), (3), (4), (5), and (18) is the intersection of the *forest polytope* on G (given by (5), (6), and (18)) and the matching polytope on $\overline{G}$ (given by (7), (8), and (9)). In the discussion to follow, we prove constructively, using Theorem 1, that this polytope's extreme points correspond to the incidence vectors of the 2-capacitated forests of G. Let us call the new polytopes P^* and Q^*, corresponding respectively to the 2-capacitated forest polytope and the polytope given by (18) and (2) through (5). We wish to show that $P^* = Q^*$. Clearly $P^* \subseteq Q^*$; we need to show that $Q^* \subseteq P^*$.

Let x be a point in Q^* and suppose that $x(E) < n-1$. To prove that $Q^* \subseteq P^*$, first we will show that for some vector $y \geq 0$, $x(E)+y(E) = n-1$, and $x+y$ is contained in $Q(=P)$; thus $x+y$ can be written as a convex combination of 2-capacitated trees, by Theorem 1. Then we will show that from the convex combination of 2-capacitated trees for $x+y$, we can construct a convex combination of 2-capacitated forests for x.

Lemma 2 Given a vector x that lies in Q^*, with $x(E) < n-1$, there exists a vector $y \geq 0$ such that $x+y$ lies in Q.

Proof We will actually show something stronger: we can select the vector $y \geq 0$ so that the strictly positive components of y correspond only to root edges (edges incident to vertex 1).

Claim 1 Let $x \in Q^*$ with $x(E) < n-1$. Then for some root edge $1u$ and $\epsilon > 0$, increasing the weight of x_{1u} by ϵ does not violate any of the constraints (2) through (4).

Actually, we will prove the contrapositive of the claim: if, for every root edge $1u$, some constraint containing x_{1u} is tight for x, then $x(E) = n-1$. Suppose that for every root edge some inequality is tight at equality. These tight inequalities are necessarily of the form $x(E(T)) \leq |T| - 1$ for some $T \ni 1$, since these are the only constraints from among (2) through (4) for which root edges have non-zero coefficients. For each root edge $1u$, we choose such a tight constraint, and let T_u denote the corresponding vertex set T. Next consider

$$A := \bigcup_{u=2}^{n} E(T_u).$$

Define

$$A_2 = E(T_2),\ A_3 = A_2 \cup E(T_3), \ldots, A_n = A_{n-1} \cup E(T_n) = A.$$

To prove that $x(A) = n-1$, we use induction on the sets A_u. The inductive hypothesis is that, for any u, if the set A_u spans the vertex set N_u, then $x(A_u) = |N_u| - 1$. Since A spans V, the inductive hypothesis for $u = n$ says that

$$x(A) = x(A_n) = |V| - 1 = n - 1.$$

The hypothesis is certainly true for $u = 2$, since $A_2 = E(T_2)$, $N_2 = T_2$, and by assumption, $x(E(T_2)) = |T_2| - 1$. Now assume that the inductive hypothesis is true for u, and consider $u + 1$. Suppose that

$$|N_u \cap T_{u+1}| = r.$$

(Note that $r \geq 1$ because node 1 is in the intersection.) Then because x is feasible,

$$x(A_u \cap E(T_{u+1})) \leq r - 1,$$

and thus

$$x(A_{u+1} \backslash A_u) \geq |T_{u+1}| - r, \tag{19}$$

since, by the definition of T_{u+1}, $x(E(T_{u+1})) = |T_{u+1}| - 1$. Now, $|N_{u+1}| = |N_u| + |T_{u+1}| - r$, and by induction

$$x(A_u) = |N_u| - 1. \tag{20}$$

Thus from inequalities (19) and (20) we have

$$\begin{aligned} x(A_{u+1}) &= x(A_{u+1} \backslash A_u) + x(A_u) \\ &\geq (|T_{u+1}| - r) + (|N_u| - 1) \\ &= |N_{u+1}| - 1, \end{aligned}$$

which completes the inductive step. Thus $x(A) = n-1$, which implies $x(E) = n-1$; therefore we have established Claim 1.

Now, a simple inductive argument will prove Lemma 2. Since $x(E) < n-1$, by Claim 1 we can increase the weight of some root edge $1u$ and remain feasible. Let ϵ_1 be the largest amount that x_{1u} can be increased without violating any constraints, and set $y^1 = \epsilon_1 \chi_{1u}$ (χ_{1u} is the indicator vector for edge $1u$). Notice that, for $x + y^1$, some constraint involving edge $1u$ is tight. Now either $x(E) + y^1(E) = n-1$, or we can increase the weight of some other root edge $1v$, $v \neq u$. As before, choose ϵ_2 as large as possible and set $y^2 = \epsilon_2 \chi_{1v}$ so that $x + y^1 + y^2$ violates no constraint. We can continue in this manner for at most $n-1$ iterations, since at each iteration we eliminate some root edge from the set of root edges whose weights can be increased. Thus eventually, for some $k \leq n-1$, by (the contrapositive of) Claim 1 we have

$$x(E) + \sum_{i=1}^{k} y^i(E) = n-1,$$

and $x + \sum_{i=1}^{k} y^i$ is feasible for Q. Setting $y = \sum_{i=1}^{k} y^i$ completes the argument. Q.E.D.

We know from Theorem 1 that we can express $x + y$ as given in Lemma 2 as a convex combination of 2-capacitated trees,

$$x + y = \sum_{i=1}^{L} \lambda_i \tau^i, \quad \sum_{i=1}^{L} \lambda_i = 1, \ \lambda \geq 0, \tag{21}$$

for some L. We wish to express x as a convex combination of 2-capacitated forests,

$$x = \sum_{i=1}^{M} \mu_i \phi^i, \quad \sum_{i=1}^{M} \mu_i = 1, \ \mu \geq 0.$$

As in the argument for Theorem 1, we assume for simplicity (so that we can temporarily clear fractions) that all λ_i and all y_{1v} are rational. Let M be chosen so that $M\lambda_i \in \mathbf{Z}$, $i = 1, \ldots, L$, and $My_{1u} \in \mathbf{Z}$, for all $u \in \overline{V}$. Consider a collection $\mathcal{C}$ of trees consisting of $M\lambda_i$ copies of τ^i, $i = 1, \ldots, L$. (Note that $|\mathcal{C}| = M$.) Consider the following procedure that transforms the 2-capacitated trees of $\mathcal{C}$ into 2-capacitated forests. We label the elements of $\mathcal{C}$ by ϕ. For each $u \in \overline{V}$, if $y_{1u} > 0$ then delete edge $1u$ from My_{1u} forests of $\mathcal{C}$ that contain edge $1u$ (*i.e.*, change ϕ_{1u} from 1 to 0).

Clearly, these deletions are possible, since $M(x+y)_{1u} \geq My_{1u}$, and $M\sum_{i=1}^{L} \lambda_i \tau^i{}_{1u} = M(x+y)_{1u}$. Furthermore, if we denote the new elements of $\mathcal{C}$ as $\phi^1, \ldots, \phi^M$, we have

$$x = \sum_{i=1}^{M} (1/M)\phi^i,$$

i.e., x is a convex combination of 2-capacitated forests. Thus we have proved the following theorem.

Theorem 2 The inequalities (2) through (5) and (18) provide a complete description of the 2-capacitated forest polytope.

As in the case of the tree polytope, it is easy to show that, for $n \geq 4$, this characterization is non-redundant.

5 Concluding Remarks

The result presented in this paper for the 2-capacitated spanning tree is easily generalized to the case when the underlying graph is directed, and the desired structure is a 2-capacitated branching [9].

In Section 1, we mentioned that Gamble and Pulleyblank have used a dual-based approach to establish the polyhedral description of the forest cover polytope [6]. An interesting open problem is whether the same approach could be used for the problem we consider in this paper. Although their result seems to parallel ours, the forest cover problem itself is algorithmically much simpler to solve than the 2-capacitated spanning tree problem. Not surprisingly, the algorithm for the forest cover problem indicates a simple tranformation of dual variables that establishes the polyhedral characterization. For the 2-capacitated spanning tree problem, there does not seem to be a straightforward dual-based approach.

Finally, we note that at least one generalization of this result is unlikely. Gamble and Pulleyblank [7] have proved that the following question is NP-hard: given a graph

$G = (V, E)$ with costs on the edges, and a subset of vertices $M \subseteq V$, find a maximum-cost acyclic subgraph whose induced subgraph on M forms a matching. The 2-capacitated spanning tree problem is a special case of this problem with $|V - M| = 1$.

Acknowledgments We would like to thank David Shmoys for valuable discussions pertaining to this work. Also, we thank Bill Pulleyblank for pointing out the more compact transformation between the 2-capacitated spanning tree problem and non-bipartite matching presented in the introduction.

References

[1] J. R. Araque, L. A. Hall, and T. L. Magnanti. Capacitated trees, capacitated routing, and associated polyhedra. Forthcoming.

[2] D. Bienstock, 1987. Private communication.

[3] J. Edmonds. Maximum matching and a polyhedron with {0, 1}-vertices. *Journal of Research of the National Bureau of Standards*, 69B:125–130, 1965.

[4] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[5] J. Edmonds. Matroid intersections. In P. L. Hammer, E. L. Johnson, and B. H. Korte, editors, *Discrete Optimization*, pages 39–49, Amsterdam, 1977. Annals of Discrete Mathematics 4, North Holland.

[6] A. B. Gamble and W. R. Pulleyblank. Forest covers and a polyhedral intersection theorem. *Math. Prog. (Series B)*, 45:49–58, 1989.

[7] A. B. Gamble and W. R. Pulleyblank, June 1989. Private communication.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., New York, 1979.

[9] L. A. Hall. *Two Topics in Discrete Optimization: Polyhedral Structure of Capacitated Trees and Approximation Algorithms for Scheduling.* PhD thesis, Mass. Inst. Tech., 1989.

[10] C. H. Papadimitriou. The complexity of the capacitated tree problem. *Networks*, 8:217–230, 1978.