

**Universitetet i Oslo
Institutt for informatikk**

The 2-path network design problem

**Geir Dahl and
Bjarne Johannessen**

**Report 292,
ISBN 82-7368-241-2**

Nov. 2, 2000



The 2-path network design problem

Geir Dahl* Bjarne Johannessen†

Nov. 2, 2000

Abstract

Given a graph with nonnegative edge weights and a set D of (terminal) node-pairs, the *2-path network design problem* asks for a minimum weight set of edges such that the induced subgraph contains a path with one or two edges connecting each pair in D . The problem is *NP*-hard. We present two integer programming models for the problem, study a relaxation and establish classes of cutting planes. A greedy approximation algorithm is developed and analyzed. Moreover, we describe a cutting plane algorithm for the problem and give some computational experiences.

Keywords: integer programming, cutting plane/facet, networks/graphs.

1 Introduction

In connection with routing in communication networks it may be important to have communication paths with very few edges in order to have small delay or high reliability. This paper treats a problem motivated by such applications. Let $G = (V, E)$ and $H = (V, D)$ be undirected graphs with the same node set and let c_e be a given nonnegative weight (cost) for each edge $e \in E$. Define $n := |V|$ and $k := |D|$. Each edge $d \in D$ is called a *demand edge*. Let $d = [s, t]$. A *2-path* for d is an st -path in G with either one or two edges. We call $F \subseteq E$, and the subgraph (V, F) , a *2-path network (for D)* if it contains a 2-path for each demand edge in D . The *2-path network design problem (2ND)* is to find a 2-path network (V, F) with minimum weight $c(F) := \sum_{e \in F} c_e$. We call G the *supply graph* and H the *demand graph*.

For two distinct nodes s and t we let $d(s, t)$ be the *distance* between them, i.e., the minimum number of edges in an st -path. The *diameter* of a graph is the maximum distance between two nodes. If $F \subseteq E$ we let $d_F(s, t)$ denote the distance between s and t in the subgraph (V, F) . In Li, McCormick & Simchi-Levi (1992) some related problems concerning edge addition problems were studied. In particular, the following result was shown in Li et al. (1992).

*University of Oslo, Dept. of Mathematics and Dept. of Informatics, P.O.Box 1080, Blindern, 0316 Oslo, Norway (geird@ifi.uio.no)

†University of Oslo, Dept. of Informatics

Theorem 1.1 *The following decision problem is NP-complete.*

(EDGE-ADD): given a graph $G_1 = (V, E_1)$ and a positive integer B , does there exist a superset $E_2 \supset E_1$ such that $|E_2 \setminus E_1| \leq B$ and the graph $G_2 = (V, E_2)$ has diameter 1 or 2.

Consider the following restricted version of (2ND), denoted by (2NDres). Let both G and H be complete graphs (on V) and assume that $c_e \in \{0, 1\}$ for each $e \in E$. Then $F \subseteq E$ is feasible in (2NDres) if and only if the diameter of the subgraph (V, F) is 1 or 2. The decision problem associated with (2NDres), which asks if there is a feasible set with weight no greater than B , is precisely the problem (EDGE-ADD) (where E_1 are the edges with cost zero). This proves the following complexity result.

Theorem 1.2 *Both (2NDres) and (2ND) are NP-hard.*

However, there is another interesting restricted version of (2ND) that is polynomially solvable.

Theorem 1.3 *Consider the restriction (2NDmatch) of (2ND) where demand edges are pairwise disjoint, i.e., D is a matching. There is a polynomial reduction of (2NDmatch) to a matching problem in a graph with $k + 1$ nodes.*

Proof. We observe that in (2NDmatch) each edge $e \in E$ lies in the 2-path of at most two demands (as D is a matching). Consider the complete graph K with node set $D \cup \{d'\}$. Let E_{ij} denote the set of (four) edges in E having one end node in common with d_i and the other endnode in common with d_j . Define a weight function w on the edge set of K as follows. For $i = 1, \dots, k$ let $w(d_i, d')$ be the minimum cost of a 2-path for d_i (in G). For $i, j \in \{1, \dots, k\}$, $i \neq j$, let $w(d_i, d_j)$ be the minimum cost of a subset of three edges chosen from the set E_{ij} (this corresponds to the situation where the 2-path for d_i and the 2-path for d_j intersect). With this weight function the (2ND)-problem becomes equivalent to finding a minimum weight set F' of edges in K such that each node d_i , for $i = 1, \dots, k$, is incident to exactly one edge in F' . This problem is a minimum weight b -matching problem (in node d' there is no degree constraint). This shows that (2NDmatch) is polynomially solvable. $\square$

Consider the special case of (2ND) where the demand graph is a star, say $D = \{[r, v] : v \in V \setminus \{r\}\}$ where r is some node in r . Thus we ask for a minimum cost edge set F such that each node $v \neq r$ is either adjacent to r or r and v have a common neighbor. This is the *2-hop spanning tree problem* which was studied in Dahl (1998). Even this problem is NP-hard as one may set up a cost function for which the simple plant location problem is (polynomially) reduced to it. When G is a wheel a complete linear description of an associated integral polytope was determined in Dahl (1998) (see also Dahl (1999) for a related study of certain stable set polytopes). For a study of minimum cost spanning tree (and Steiner tree) problems with more general hop constraints we refer to Gouveia (1996).

Consider the special case of (2ND) where $c_e = 1$ for each $e \in E$. Then we can easily solve the problem as follows. Let $V_1, \dots, V_t$ be the (connected) components of the demand graph, and select a node $v_i \in V_i$ for $i = 1, \dots, t$. Let F_i be the “substar” given by $F_i = \{[v_i, w] : w \in V_i \setminus \{v_i\}\}$ for each i and let F^* be the union of these t node-disjoint substars. Then F^* is an optimal solution of (2ND). To see this, consider a 2-path network F . Then each V_i must be a subset of some component of (V, F) (“connected demands implies connected 2-paths”). This implies that $|F| \geq \sum_{i=1}^t (|V_i| - 1) = |F^*|$ which proves the optimality of F^* under the unit cost assumption. Thus, in (2ND) instances where costs are nearly equal we may expect an optimal solution to contain several substars.

For graph theory and polyhedral theory used in this paper, see Nemhauser & Wolsey (1988) and Schrijver (1986). For disjoint node sets S and T we let $[S, T]$ denote the set of edges joining a node in S and a node in T . We write $[s, T]$ instead of $[\{s\}, T]$. For each demand edge $d \in D$ we let e_d denote the (unique) edge in E that is parallel to d . A *root* of an inequality $\mathbf{c}^T \mathbf{x} \geq \alpha$ is a point $\mathbf{x}'$ satisfying $\mathbf{c}^T \mathbf{x}' = \alpha$.

In the remaining part of the paper we first introduce two integer programming formulations of (2ND) (section 2). An approximation algorithm for the problem is studied in section 3 while section 4 contains a study of related polyhedra and cutting planes for the (2ND) problem. A cutting plane algorithm is presented in section 5 along with computational results and experiences.

2 Integer programming formulations

We give two different formulations of (2ND) as an integer linear programming problem. For notational simplicity we shall assume throughout that the supply graph is complete (see the remark below).

Let s and t be distinct nodes in G and define a *2-star* as a subset of E of the form

$$T(S_1, S_2) = [s, S_2] \cup [t, S_1] \quad (1)$$

where $S_1 \cup S_2 = V$, $S_1 \cap S_2 = \emptyset$ and $s \in S_1$, $t \in S_2$. We say that $T(S_1, S_2)$ is a 2-star *for* the edge $[s, t]$. In particular, the stars $\delta(s)$ (consisting of all edges incident to s) and $\delta(t)$ are both 2-stars. Let $\mathcal{T}_{st}$ denote the set of all 2-stars for $\{s, t\}$ and let $\mathcal{T} = \cup_{[s, t] \in D} \mathcal{T}_{st}$ be the set of 2-stars associated with the demand edges.

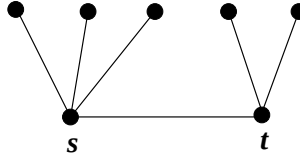


Figure 1: A 2-star.

Lemma 2.1 *Let s and t be two distinct nodes in G and let $F \subseteq E$. Then (V, F) contains a 2-path between s and t if and only if $F \cap T$ is nonempty for every $T \in \mathcal{T}_{st}$.*

Proof. Each 2-path in (V, F) is either the single edge $[s, t]$ or of the form $[s, v], [v, t]$ for some $v \in V \setminus \{s, t\}$ provided that these edges lie in F . Let $E_1 \subseteq F$ be the union of these edge sets, and define the corresponding subgraph $G_1 = (V, E_1)$. Note that in G_1 all st -paths are 2-paths. By Menger's theorem (or the max-flow min-cut theorem, see e.g. Nemhauser & Wolsey (1988)) G_1 contains an st -path if and only if each st -cut is nonempty. But st -cuts in G_1 and 2-stars coincide, and the desired result follows. $\square$

Remark: We have assumed that G is complete, but for a general graph G there is an appropriate definition of 2-star which works in the sense that Lemma 2.1 still holds. Let E' be the set of edges that lie in some 2-path (for some demand) and define a 2-star in G to be the intersection of E' and a 2-star in the associated complete graph.

Due to Lemma 2.1, the following integer linear program is a valid formulation of the (2ND) problem:

$$\begin{aligned}
& \min && \sum_{e \in E} c_e x_e \\
& \text{subject to} && \\
& \text{(i)} && \sum_{e \in T} x_e \geq 1 \quad \text{for all } T \in \mathcal{T}; \\
& \text{(ii)} && x_e \in \{0, 1\} \quad \text{for all } e \in E.
\end{aligned} \tag{2}$$

We call this model the *2-star model*. The variable x_e indicates if edge e is in the chosen 2-path network. Note here that the number of *2-star inequalities* (2)(i) grows exponentially in n .

Another model of (2ND) is obtained by introducing additional variables reflecting the 2-path chosen for each demand edge. Let $\mathcal{P}_d$ be the set of all 2-paths for demand $d \in D$. The following model is called the *path model*:

$$\begin{aligned}
& \min && \sum_{e \in E} c_e x_e \\
& \text{subject to} && \\
& \text{(i)} && \sum_{P \in \mathcal{P}_d} z_P = 1 \quad \text{for all } d \in D; \\
& \text{(ii)} && z_P \leq x_e \quad \text{for all } e \in P, P \in \mathcal{P}_d, d \in D; \\
& \text{(iii)} && x_e, z_P \in \{0, 1\} \quad \text{for all } e \in P, P \in \mathcal{P}_d, d \in D.
\end{aligned} \tag{3}$$

In a feasible solution one selects exactly one candidate 2-path for each demand as well as all edges in the support of these chosen paths. A feature of this model is the flexibility of allowing “routing cost” by adding the term $\sum_P \theta_P z_P$ to the objective function.

The path model has $k(n-1)$ additional variables compared to the 2-star model. On the other hand, the path model has $kn(n-1)^2/2 + k$ constraints

(plus simple bounds) while the corresponding number in the 2-star model is $k \cdot 2^{n-2}$. The path model is therefore a compact extended formulation (the number of variables and constraints grow polynomially in n and k).

Let LP_p (resp. LP_{2s}) be the feasible polyhedron when integrality is dropped (and replaced by lower and upper bounds of 0 and 1) in the path model (resp. the 2-star model).

Proposition 2.2 *The projection of LP_p into the space of the x_e -variables is LP_{2s} . Therefore the LP relaxations of the path model and the 2-star model have the same optimal value.*

Proof. Let $(\mathbf{x}, \mathbf{z}) \in LP_p$ where $\mathbf{x} = (x_e : e \in E)$ and $\mathbf{z}$ is the vector of all the path variables z_P . Let $[s, t] \in D$ and consider the subgraph G' of G induced by the edge set $\delta(s) \cup \delta(t)$. It follows from (3)(i) and (ii) that the corresponding directed network G' equipped with nonnegative capacities $\mathbf{x}_e$ allows a flow of value 1 from s to t . By the max-flow min-cut theorem this implies that $\mathbf{x}$ must satisfy all the 2-star constraints associated with the demand $[s, t]$. This proves that $\mathbf{x} \in LP_{2s}$. Thus the projection of LP_p into the space of the x_e -variables is contained in LP_{2s} . The converse containment is proved similarly. $\square$

We remark that it is possible to transform the path model into a set packing problem (by complementing edge variables, confer the constraints (3)(ii)). This opens up for transforming strong inequalities known for set packing polytopes to the present situation although we do not pursue this idea here.

3 A fast approximation algorithm

In this section we suggest and analyze an approximation algorithm for the (2ND) problem. The main idea is to approximately solve the dual of the linear relaxation of the 2-star model and generate a corresponding primal feasible solution.

The LP relaxation of the 2-star model (2) has the following dual problem:

$$\begin{aligned}
& \max && \sum_{T \in \mathcal{T}} y_T \\
& \text{subject to} && \\
& \text{(i)} && \sum_{T \in \mathcal{T}: e \in T} y_T \leq c_e \quad \text{for all } e \in E; \\
& \text{(ii)} && y_T \geq 0 \quad \text{for all } T \in \mathcal{T}.
\end{aligned} \tag{4}$$

In this problem there is a nonnegative variable y_T for each 2-star $T \in \mathcal{T}$. Each constraint (4)(i) may be seen as a packing constraint. Our approximation algorithm for (2ND) solves (4) approximately and constructs a primal feasible solution along the way. The algorithm uses similar ideas as presented in Goemans & Williamson (1995) for a wide class of constrained forest problems. Since feasible and optimal solutions in (2ND) may not be forests, the approximation algorithm of Goemans & Williamson (1995) can not be applied directly to (2ND). However, the general dual ascent scheme provides a framework for our

techniques. See Nemhauser & Wolsey (1988) for a general discussion of dual ascent algorithms.

The *Greedy* algorithm may be described as follows. We maintain an *active pool* of node sets consisting of two node sets for each *active demand*. For each active demand edge $[s, t] \in D$ the pool contains two subsets S_s and S_t of V satisfying

$$s \in S_s, t \notin S_s \text{ and } t \in S_t, s \notin S_t. \quad (5)$$

We start with $S_s = \{s\}$ and $S_t = \{t\}$ and all demands are active. Then gradually demands are made inactive and then the corresponding node sets are removed from the active pool. The algorithm also constructs an edge set F , starting with the empty set, and the final primal solution $\bar{F}$ is extracted from F .

The algorithm performs at most $|E|$ iterations. In each iteration the following is done:

1. an edge is added to F ;
2. at least one of the sets in the active pool is expanded by a node;
3. those demands for which F contains a 2-path are made inactive.

The algorithm proceeds until all demands are inactive. Then F is clearly feasible and the final stage is to extract the output solution $\bar{F}$ from F by deleting edges without violating feasibility.

To give more details, consider the start of an iteration. Let $\bar{c}_e$ for each $e \in E$ be the *reduced edge costs*. These numbers are known from the previous iteration. We start the algorithm with $\bar{c}_e = c_e$ for each e . For each active demand $d = [s, t]$ our active pool contains the two sets S_s and S_t and we define the two corresponding *active 2-stars*

$$T(S_s, V \setminus S_s) \text{ and } T(V \setminus S_t, S_t).$$

These two sets are called *complementary 2-stars*. Note that the complementary 2-stars form a partition of the edge set $\delta(s) \cup \delta(t) \setminus \{[s, t]\}$, and this property is maintained as long as $[s, t]$ is active. For each edge $e \in E$ let m_e denote the number of active 2-stars containing e and define

$$\epsilon^* = \max\{\epsilon : \epsilon \cdot m_e \leq \bar{c}_e \text{ for all } e \in E\}. \quad (6)$$

For each 2-star T we have a variable y_T in the dual LP problem (4) and the initial value is $y_T = 0$. Implicitly, after the calculation of ϵ^* , we update y_T for each active demand by setting $y_T := y_T + \epsilon^*$. From our choice of ϵ^* we see that the resulting vector of dual variables $\mathbf{y}$ is feasible in (4). We also update the reduced cost by $\bar{c}_e := \bar{c}_e - \epsilon^* \cdot m_e$ for each $e \in E$ and let $F := F \cup F_1$ where F_1 denotes the new set of edges for which the reduced cost just became zero. Finally, for each node set S in the active pool which contains exactly one endnode of an edge in F_1 we add the other endnode to S .

It is clear that at most $|E|$ iterations are required and that the final solution $\mathbf{y}$ is feasible in (4). Thus, by linear programming (weak) duality, a lower bound on the optimal value z of (2) is $\bar{z}_l := \sum_T y_T$. The primal solution $\bar{F}$ constructed in

the final stage gives an upper bound on z . Thus when the algorithm terminates, we have a performance guarantee in terms of these two bounds.

We now analyze the performance of the *Greedy* algorithm. Let $d_H(v)$ denote the degree of a node v in the demand graph H . Recall that $k = |D|$ is the number of demand edges.

Theorem 3.1 *Let z^* be the optimal value in (2ND) and let $\bar{z}_l$ and $\bar{F}$ be the lower bound and primal solution, respectively, that are produced by the Greedy algorithm. Then*

$$\bar{z}_l \leq z^* \leq \mathbf{c}(\bar{F}) \leq \gamma \cdot \bar{z}_l \quad (7)$$

where $\gamma = \frac{1}{2k} \sum_{v \in V} (d_H(v))^2 + \max_{v \in V} d_H(v)$. Therefore Greedy is a γ -approximation algorithm for the (2ND) problem. In particular, for the (2NDmatch) problem (D is a matching) we have $\gamma = 2$.

Proof. Let $M := \max_{v \in V} d_H(v)$. Each feasible solution F in (2) may be written $F = \cup_{d \in D} F_d$ where F_d is a 2-path for demand d . The important step is to prove the following:

Claim: Let $D' \subseteq D$ and let the set $\mathcal{T}'$ consist of pairs of complementary 2-stars, one pair associated with each demand edge in D' . Then

$$(*_1) \quad \sum_{T \in \mathcal{T}'} |F \cap T| \leq 2k \cdot \gamma.$$

To prove this, consider a demand edge $d = [u_d, v_d]$. Recall that if an edge e lies in a 2-star T associated with some demand edge d , then e is incident to one of the endnodes of d . Note also that a pair of complementary 2-stars for d only intersect in one edge (namely e_d). If $|F_d| = 1$ then F_d can only intersect the 2-stars associated with demand edges incident to u_d or v_d , so in this case we conclude that $\sum_{T \in \mathcal{T}'} |F_d \cap T| \leq d_H(u_d) + d_H(v_d)$. If $|F_d| = 2$, say $F_d = \{[u_d, w], [w, v_d]\}$, then F_d may also intersect the 2-stars associated with demands incident to w , and there are at most $M = \max_{v \in V} d_H(v)$ of these, so we have $\sum_{T \in \mathcal{T}'} |F_d \cap T| \leq d_H(u_d) + d_H(v_d) + 2M$. From these estimates we calculate

$$\begin{aligned} \sum_{T \in \mathcal{T}'} |F \cap T| &= \sum_T |(\cup_{d \in D'} F_d) \cap T| = \sum_T |\cup_d (F_d \cap T)| \\ &\leq \sum_T \sum_d |F_d \cap T| = \sum_d \sum_T |F_d \cap T| \\ &\leq \sum_d (d_H(u_d) + d_H(v_d) + 2M) \\ &= \sum_v (d_H(v))^2 + 2kM \leq 2k \cdot \gamma \end{aligned}$$

which proves the claim.

Consider now the *Greedy* algorithm and let $\bar{F}$ be the primal solution produced. The algorithm also finds dual variables y_T for each 2-star T and, moreover, we have $c_e = \sum_{T: e \in T} y_T$ for each $e \in \bar{F}$. In fact, this equality holds for all edges added to the set $\bar{F}$ during the algorithm and $\bar{F} \subseteq F$. Therefore

$$\mathbf{c}(\bar{F}) = \sum_{e \in \bar{F}} c_e = \sum_{e \in \bar{F}} \sum_{T: e \in T} y_T = \sum_T y_T |\bar{F} \cap T|$$

so (7) will follow if we prove that

$$(*_2) \quad \sum_T y_T |\bar{F} \cap T| \leq \gamma \cdot \sum_T y_T.$$

Clearly, $(*_2)$ holds initially as $y_T = 0$ for each T . During the first iteration we set $y_T = \epsilon$ for each active 2-star T . Thus, the left hand side of $(*_2)$ becomes $\epsilon \sum_T |\bar{F} \cap T|$ and the right hand side becomes $\epsilon \gamma \cdot 2k$. But then the inequality in $(*_2)$ holds due to $(*_1)$. Furthermore, in each of the other iterations of the algorithm the increase of the left hand side is no greater than the increase on the right hand side; this is again due to the claim and the inequality $(*_1)$ above. This proves that $(*_2)$ holds when the algorithm terminates and therefore (7) holds. Finally, if D is a matching then $d_H(v) \leq 1$ for each $v \in V$, $M = 1$ and this gives $\gamma = 2$. $\square$

Some remarks on the quality of the bound in Theorem 3.1 are in order. First we observe that 2 is the best possible bound in (7) when D is a matching. This is seen from the following example. Let $s_1, \dots, s_k, t_1, \dots, t_k$ be distinct nodes and let D consist of the demand edges $[s_i, t_i]$ for $i \leq k$. Let F consist of the edges $[s_i, s_{i+1}]$ and $[s_{i+1}, t_i]$ for $i \leq k$ where $s_{k+1} := s_1$. Define edge costs in G by $c_e = 1$ for each $e \in F$ and $c_e = 3$ for all other edges. Then, after one iteration, the *Greedy* algorithm will give F as output and $y_T = 1/2$ for all stars of the endnodes of demand edges. Thus we obtain the lower bound $\bar{z}_l = 2k \cdot 1/2 = k$ and the upper bound $\mathbf{c}(F) = 2k$ and the performance bound 2 is best possible. Note that the algorithm in this example does find an optimal solution, but it is not able to verify this because the lower bound $\bar{z}_l$ is too low. Based on extensive computational tests we have seen the same pattern: the quality of the solutions found by *Greedy* may be quite good, but the lower bounds are not, see section 5. Note that γ increases quadratically with the degrees of the nodes in the demand graph, so when there are nodes of high degree, γ becomes large. Finally, we remark that there is reason to believe that finding good approximation algorithms for (2ND) is difficult (confer the discussion in connection with Theorem 1.1 in Li et al. (1992)).

4 Related polyhedra and cutting planes

In this section we present different ways of strengthening the linear relaxations associated with the path and 2-star models (see section 2). First we study some

properties of the 2-star relaxation. As usual we assume that the supply graph G is complete.

For two disjoint demand edges $d_i = [s_i, t_i]$ and $d_j = [s_j, t_j]$ we define $E_{i,j} := \{[s_i, s_j], [s_i, t_j], [t_i, s_j], [t_i, t_j]\}$, i.e., the set of supply edges “between” d_i and d_j . A subset F obtained by removing at most one edge from $E_{i,j}$ is called a *connection* for d_i and d_j . Thus, if $|F| = 3$ this connection contains a 2-path for d_i and a 2-path for d_j . If $F = E_{i,j}$, then F contains two 2-paths for each demand.

4.1 A result concerning the 2-star relaxation

We investigate the fractional vertices of LP_{2s} , the feasible polyhedron in the 2-star relaxation. Later we introduce strong cutting planes that cut off such vertices.

Let $d_{j_1}, \dots, d_{j_t}$ be an odd number of pairwise disjoint demand edges. A *cycle-connection* is a union of connections $F_1, \dots, F_t$ where each F_r is a connection for d_{j_r} and $d_{j_{r+1}}$ (where $j_{t+1} = j_1$). For $\mathbf{x} \in LP_{2s}$ we let $G^f(\mathbf{x})$ denote the subgraph of G induced by the set $\{e \in E : 0 < x_e < 1\}$. By a *dumbbell* we mean a graph obtained from a tree T by adding an edge between two nonadjacent nodes in T . Thus, a dumbbell has a unique cycle and we say that the dumbbell is *odd* if this cycle is odd.

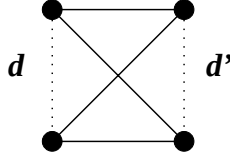


Figure 2: A connection.

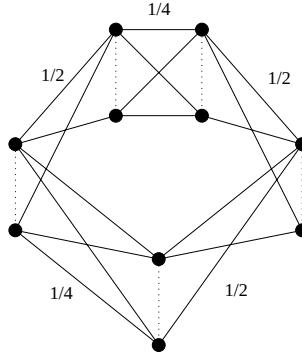


Figure 3: A cycle-connection.

The following result characterizes the vertices of LP_{2s} in the case when the demand edges are pairwise disjoint. Recall from section 1 that this corresponds to the special case (2NDmatch) which may be solved efficiently as a matching problem in a graph with $k + 1$ nodes. The proof is given in the appendix.

Theorem 4.1 *Assume that the demand set D is a matching. Let $\mathbf{x} \in LP_{2s}$. Then $\mathbf{x}$ is a vertex of LP_{2s} if and only if every component of $G^f(\mathbf{x})$ is either (i) a connection of cardinality 4 with all its variables being $1/2$, or (ii) an odd cycle-connection where all variables are $1/2$ in its connections of cardinality 3 and they are $1/4$ in its connections of cardinality 4. In particular, each vertex $\mathbf{x}$ satisfies $x_e \in \{0, \frac{1}{4}, \frac{1}{2}, 1\}$ for all $e \in E$.*

For general demand graphs one may determine large classes of vertices of LP_{2s} (but not all vertices) based on the structural results of Theorem 4.1, but we omit this here. A main consequence of the theorem is that it leads to large classes of strong cutting planes for the (2ND) problem in general (for all demand graph), see below.

4.2 The demand-pair inequalities

We now turn to the question of strengthening the 2-star formulation by finding additional (strong) valid inequalities. The inequalities are derived by considering different types of demand graphs H . Let P_{2s} denote the convex hull of feasible (integral) vectors in the 2-star model. (The dependency of G and H is suppressed and should be clear from the context below.) All the inequalities discussed are of the form $\sum_{e \in E} a_e x_e \geq \alpha$ where $\mathbf{a}$ is nonnegative and $\alpha > 0$. Thus, they only involve the edge variables x_e , $e \in E$. Clearly, if such an inequality is valid for P_{2s} , then it is also valid in the path model. We shall therefore restrict the attention to polyhedral properties for the polytope P_{2s} . This polytope is full-dimensional (except in the trivial case when G has only two nodes).

Let $d_i = [s_i, t_i]$ and $d_j = [s_j, t_j]$ be two disjoint demand edges in H . We now choose two edge sets L_1 and L_2 as follows. Let $L_1 \subset E$ consist of $[s_i, t_i]$, $[s_j, t_j]$ as well as three edges chosen from the set $E_{i,j} := \{[s_i, s_j], [s_i, t_j], [t_i, s_j], [t_i, t_j]\}$. For each $v \notin \{s_i, t_i, s_j, t_j\}$ choose (i) one of the two edges $[v, s_i]$ and $[v, t_i]$, and (ii) one of the two edges $[v, s_j]$ and $[v, t_j]$, and let L_2 be the set of these chosen edges. The inequality

$$\mathbf{x}(L_1 \cup L_2) \geq 2 \tag{8}$$

is called a *demand-pair inequality of type I*. There are 2^{2n-6} different such inequalities for each pair of disjoint demands.

Proposition 4.2 *Each demand-pair inequality of type I is valid for P_{2s} . It defines a facet of P_{2s} provided that each demand edge $d \in D \setminus \{d_i, d_j\}$ does not intersect d_i or d_j .*

Proof. We first consider the case when $V = V_0 := \{s_i, t_i, s_j, t_j\}$ and $D = \{d_i, d_j\}$ given by $d_i = [s_i, t_i]$ and $d_j = [s_j, t_j]$. Then the inequality (8) is easily

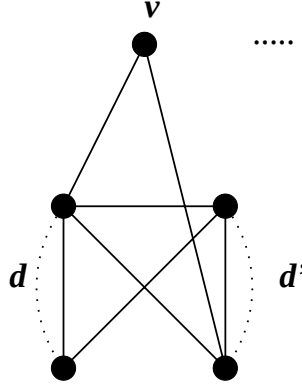


Figure 4: Demand-pair inequality of type I.

seen to be valid and facet defining. Assume that V has five nodes, say node v in addition to the four mentioned nodes. Let L_1 consist of $[s_i, t_i]$, $[s_j, t_j]$ and three edges chosen from $E_{i,j}$. Consider the lifted inequality

$$\mathbf{x}(L_1) + \sum_{u \in E_{i,j}} a_{uv} x_{uv} \geq 2.$$

This inequality is valid for P_{2s} (when G has node set V) provided that (i) either $a_{s_i,v} \geq 1$ or $a_{t_i,v} \geq 1$, and (ii) $a_{s_j,v} \geq 1$ or $a_{t_j,v} \geq 1$. By considering the best choice of these coefficients in both (i) and (ii) we obtain the demand-pair inequalities of type I. Thus these inequalities are valid for P_{2s} and, moreover, they are facet defining due to general lifting theory (see Nemhauser & Wolsey (1988)). More generally, if V has more than five nodes, we can add one node at the time in the lifting procedure. This proves that every demand-pair inequality of type I is valid and facet defining for P_{2s} when $D = \{d_i, d_j\}$. We indicate how to prove the final statement of the proposition. Consider a feasible 2-path network F for $\{d_i, d_j\}$ such that χ^F satisfies (8) with equality. Then the incidence vector of $F \cup \{e_d : d \in D \setminus \{d_i, d_j\}\}$ is a feasible 2-path network for D which satisfies (8) with equality. Using these and related roots one may prove the last statement. $\square$

Example. Consider the graph $G = K_4$ with four nodes s_1, t_1, s_2, t_2 and two disjoint demands $d_1 = [s_1, t_1]$ and $d_2 = [s_2, t_2]$. The point $\mathbf{x}'$ where $x'_e = 1/2$ for each $e \in E_{1,2}$ (and the other components are zero) is a vertex of the polytope LP_{2s} , see Theorem 4.1. This fractional vertex is cut off by e.g. the demand-pair inequality $x_{s_1,s_2} + x_{s_1,t_2} + x_{t_1,s_2} \geq 2$ as inserting $\mathbf{x}'$ gives $3/2$ on the left-hand side.

Thus, demand-pair inequalities of type I cut off fractional vertices where the “fractional graph” $G^f(\mathbf{x})$ contains connections (confer Theorem 4.1). We remark that one can show that (8) is facet defining also under weaker assumptions than in Proposition 4.2.

Another type of demand-pair inequalities is introduced next. Let again $d_i = [s_i, t_i]$ and $d_j = [s_j, t_j]$ be two disjoint demand edges in H . Let M_1 consist of (i) all edges in $E_{i,j}$ plus $[s_j, t_j]$ and (ii) for each $v \notin \{s_i, t_i, s_j, t_j\}$ choose one of the two edges $[v, s_j]$ and $[v, t_j]$. Moreover, let M_2 consist of the edge $[s_i, t_i]$ and one of the edges $[v, s_i]$ and $[v, t_i]$ for each $v \notin \{s_i, t_i, s_j, t_j\}$. The inequality

$$\mathbf{x}(M_1) + 2\mathbf{x}(M_2) \geq 3 \quad (9)$$

is called a *demand-pair inequality of type II*. (A similar inequality is obtained by changing the role of d_i and d_j).

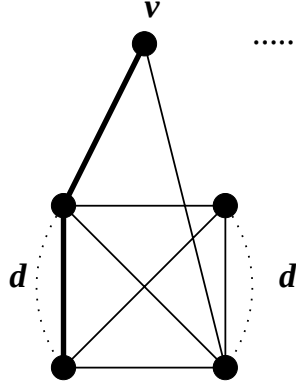


Figure 5: Demand-pair inequality of type II.

There are 2^{2n-7} demand-pair inequalities of type II for each pair of disjoint demands. Using techniques as in the proof of Proposition 4.2 one may establish the following result.

Proposition 4.3 *Each demand-pair inequality of type II is valid for P_{2s} . It defines a facet of P_{2s} provided that each demand edge $d \in D \setminus \{d_i, d_j\}$ does not intersect d_i or d_j .*

Demand-pair inequalities of type II, similar to those of type I, cut off fractional vertices where $G^f(\mathbf{x})$ contains connections.

4.3 The cycle-connection inequalities

Let D_1 be a subset of D consisting of an odd number of pairwise disjoint demand edges (so D_1 is a matching). Let $|D_1| = 2p+1$ where $p \geq 1$. Define V_1 as the set of endnodes of edges in D_1 . Let N_1 contain all edges in E with both endnodes in V_1 except the edges e_d for $d \in D_1$ (recall that e_d is the demand edge parallel to d). Let $N_2 \subset E$ consist of each e_d for $d \in D_1$ and, in addition, one of the two edges $[v, s]$ and $[v, t]$ for each $v \in V \setminus V_1$ and each $[s, t] \in D_1$. We call the inequality

$$2\mathbf{x}(N_2) + \mathbf{x}(N_1) \geq 3p + 2 \quad (10)$$

a cycle-connection inequality.

Lemma 4.4 *Each cycle-connection inequality is valid for P_{2s} .*

Proof. Let $\mathbf{c}^T \mathbf{x}$ denote the left-hand side of the inequality (10) and consider χ^F where F is a 2-path network. Then F contains the union of 2-paths $P_1, \dots, P_{2p+1}$, one for each $d \in D_1$. Note that $\mathbf{c}^T \chi^{P_i} = 2$ for $i = 1, \dots, 2p+1$. As the demand edges in D_1 are pairwise disjoint, each edge $e \in E$ can lie in at most two of the 2-paths $P_1, \dots, P_{2p+1}$. This implies that the lowest possible value of $\mathbf{c}^T \chi^F$ arises when demand edges are matched by using connections of size 3 for pairs of demand edges. (This means that the cost of the two 2-paths becomes 3, compared to 4 if they were disjoint). But, as the number of demands is odd ($2s+1$) there will be one demand unmatched and therefore the minimum value becomes $\mathbf{c}(F) = 3p + 2$ as desired. $\square$

Cycle-connection inequalities cut off fractional vertices of LP_{2s} for which the “fractional graph” $G^f(\mathbf{x})$ contains odd cycle-connections (see Theorem 4.1). Consider a vertex $\mathbf{x}$ of LP_{2s} and assume that $G^f(\mathbf{x})$ contains an odd cycle-connection C associated with the demand set D_1 . Let $|D_1| = 2p + 1$ and consider the cycle-connection inequality (10) corresponding to D_1 . Let r denote the number of connections in C that are of size 3. Note that $x_e = 1/2$ or $x_e = 1/4$ according to the cardinality of the connection in which an edge e lies. We calculate $2\mathbf{x}(F_2) + \mathbf{x}(F_1) = r(3 \cdot \frac{1}{2}) + (2p + 1 - r)(4 \cdot \frac{1}{4}) = r/2 + 2p + 1$ which ranges from $2p + 1$ to $3p + 3/2$ depending on r . Thus, independent of r , we conclude that $\mathbf{x}$ will be cut off by the cycle-connection inequality (10).

Proposition 4.5 *Assume that $D_1 \subseteq D$ consists of an odd number of demands such that each $d \in D_1$ is disjoint from every other demand in D . Then the cycle-connection inequality (10) defines a facet of P_{2s} .*

Proof. We just sketch the proof. As our previous facet proofs the demands outside D_1 cause no problems (due to disjointness assumption) so we concentrate on the case $D_1 = D$. A root of the cycle-connection inequality (10) was described in Lemma 4.4: match pairs of demands in D_1 using a connection for each and the single remaining demand $d = [s, t]$ is “routed” on e_d . Alternative roots are obtained by replacing e_d by $[s, v]$ and $[v, t]$ for a node $v \notin V_1$. Moreover, other roots are found by varying the matching of demands (in any way as G is complete). It is not difficult to show that these roots span a hyperplane in $\mathbb{R}^E$ and the desired result follows. $\square$

We remark that one can show that (10) is nonredundant under weaker assumptions than those given in Proposition 4.5.

4.4 The spanning tree inequalities

We now consider some inequalities that are derived from demand graphs where demands may intersect.

Let S be a node subset with the property that the demand subgraph $H[S]$ is connected. For each node $v \in V \setminus S$ choose an edge $e(v) \in [S, v]$ (i.e., and

edge between v and some node in S) and let $T := \{e(v) : v \in V \setminus S\}$. Let $\gamma(S)$ denote the set of edges with both endnodes in S . The inequality

$$\mathbf{x}(\gamma(S) \cup \delta(S) \setminus T) \geq |S| - 1 \quad (11)$$

is called a *spanning tree inequality*.

Lemma 4.6 *For each node set S and edge set T as above the spanning tree inequality (11) is valid for P_{2s} .*

Proof. Since $H[S]$ is connected, any 2-path network (V, F) contains all nodes in S in the same component. Let this component have node set V' where $S \subseteq V'$. Thus, F must contain at least $|V'| - 1$ edges and, among these, at most $|V' \setminus S|$ edges lie in T (because no node outside S is incident to more than one edge in T , by construction of T). Thus, the number of edges in $\gamma(S) \cup \delta(S)$ is at least

$$|V'| - 1 - |V' \setminus S| = |S| - 1.$$

which proves the validity. $\square$

Consider the special case when S consists of two nodes, say u and v . By assumption $[u, v]$ is a demand edge and we see that the edge set $\gamma(S) \cup \delta(S) \setminus T$ coincides with the 2-star defined by $[u, v]$. Thus the 2-star inequalities are special cases of spanning tree inequalities.

We briefly discuss which of the spanning tree inequalities that are nonredundant for P_{2s} . The answer depends on the demand graph H . A simple situation where the spanning tree inequality is nonredundant, is described in the following result.

Proposition 4.7 *Assume that S is a node set containing a node s such that the demand graph contains the edges $[s, v]$ for $v \in S \setminus \{s\}$ and that every other demand edge has both endnodes outside S . Let $e(v) = [v, s]$ for each $v \in V \setminus S$. Then the spanning tree inequality (11) defines a facet of P_{2s} .*

Proof. Recall that P_{2s} is full-dimensional. Thus, to prove that (11) defines a facet it suffices to show that: if each root of (11) satisfies $\sum_{e \in E} b_e x_e = \beta$, where $\sum_{e \in E} b_e x_e \leq \beta$ is valid for P_{2s} , then $\sum_{e \in E} b_e x_e \leq \beta$ is a positive multiple of (11). For each $e \in E$ having coefficient zero in (11) one can find a root of (11) which avoids e ; this implies that $b_e = 0$. Let E' denote the set of edges $e \in E$ with both endnodes outside S . Consider the root $F = (\delta(s) \cap \gamma(S)) \cup E'$ and let $F' = (F \setminus \{[s, v]\}) \cup \{[v, v']\}$ for some $v, v' \in S \setminus \{s\}$ (where $v \neq v'$). Then F' is also feasible and, moreover, it is a root of (11) (here we exploit the star-structure of the demands in $\gamma(S)$). This implies that b_e is the same number for all edges $e \in \gamma(S)$. Finally, consider again the root F and modify it by replacing, for some $v \in S \setminus \{s\}$ and some $v' \in V \setminus S$, the edge $[s, v]$ by $[v, v']$. The new solution is also a root so $b_{sv} = b_{vv'}$. Altogether we have now shown that $\sum_{e \in E} b_e x_e \leq \beta$ is a positive multiple of (11) so we conclude that the spanning tree inequality (11) defines a facet of P_{2s} . $\square$

There are weaker conditions under which the spanning tree inequality is still nonredundant (e.g., when H is a path), but we do not go further into these matters here.

5 Computations

In this section we describe a cutting plane algorithm for (2ND) which is based on the results in the previous section. Moreover, we report computational results for this algorithm where it is compared to the greedy heuristic described in section 3.

5.1 A cutting plane algorithm

We describe a branch-and-cut algorithm for solving (2ND). Our main goal was to see the effect of adding some of the cutting planes introduced in section 4, so the implementation was kept as simple as possible.

We based the algorithm on the path model although that increased the number of variables considerably. The advantage of using the path model is that there was no need to separate for the 2-star constraints (confer the constraints $\sum_{P \in \mathcal{P}_d} z_P = 1$ and Proposition 2.2). We used a branch-and-cut approach which basically is a branch-and-bound algorithm except that in the node of the branch-and-bound tree we solve the LP using a cutting plane algorithm. That is, we add cutting planes using separation procedures until no more such violated cutting planes can be found. Note that all these cutting planes are globally valid so they also hold for all other node in the branch-and-bound tree.

We implemented separation algorithms for the demand-pair inequalities of both type I and type II and for a subclass of the spanning tree inequalities.

The separation algorithm for the demand-pair inequalities (of both types) goes as follows. Assume $\mathbf{x} \in \mathbb{R}^E$ is a given vector (which, together with a path vector $\mathbf{z}$, is feasible in the path model). We go through each pair of demands and for each pair $d_i = [u_i, v_i]$, $d_j = [u_j, v_j]$ of demands the following is done:

- *Look for violated inequality of type I:* Let F consist of the three edges in $E_{i,j}$ obtained by removing the edge e with maximum value x_e . For each node v which is not an endnode of d_i or d_j do
 - a) choose the edge e among $[u_i, v]$ and $[v_i, v]$ with the smaller value x_e . Add this edge to F .
 - b) choose the edge e among $[u_j, v]$ and $[v_j, v]$ with the smaller value x_e . Add this edge to F .

If $x(F) < 2$, a violated inequality has been found.

- *Look for violated inequality of type II:* Let $F_1 := E_{i,j}$ and $F_2 := \{d_i\}$ (or d_j). For each node v which is not an endnode of d_i or d_j do
 - a) choose the edge e among $[u_i, v]$ and $[v_i, v]$ with the smaller value x_e . Add this edge to F_2 .
 - b) choose the edge e among $[u_j, v]$ and $[v_j, v]$ with the smaller value x_e . Add this edge to F_1 .

If $x(F_1) + 2\mathbf{x}(F_2) < 3$, a violated inequality has been found.

It is easy to see that this separation algorithm is polynomial. Thus, the separation problem for the demand-pair inequalities can be solved polynomially. Due to the equivalence of separation and optimization (see Grötschel, Lovász & Schrijver (1988)), this implies that we can solve the linear relaxation given by all 2-star inequalities and all demand-pair inequalities in polynomial time.

We also included separation for some of the spanning tree inequalities by checking node sets S with three nodes. The branch-and-cut tree was traversed using breadth-first-search. The selection of branching variable was done by picking one that was as near $1/2$ as possible.

We conclude this section with a remark concerning the path model given in (3) where we replace the equalities (i) by $\sum_{P \in \mathcal{P}_d} z_P \geq 1$. If we relax these constraints in a Lagrangian way, we obtain Lagrangian subproblems that may be transformed into minimum st -cut problems in a directed graph having a node for each variable in (3). The bound obtained by solving the Lagrangian dual (using e.g. the subgradient algorithm) coincides with the value of the LP relaxation of (3). Still, the Lagrangian approach may be of interest as the subproblems may be solved very fast.

5.2 Computational results

This subsection contains an empirical study of our algorithms, the cutting plane algorithm given above and the greedy algorithm described in section 3. In order to test the algorithms we generated different classes of test problems as explained below.

The tables contain the following information:

v_2	optimal value of LP relaxation of path model
v_1	optimal value of LP relaxation with additional cutting planes
v_0	optimal value of integer program
GAP1	LP relaxation gap: $(v_2 - v_0)/v_0 \cdot 100\%$
GAP2	cutting plane gap: $(v_1 - v_0)/v_0 \cdot 100\%$
CUT	total number of cutting planes added
NOD	number of nodes in branch-and-bound tree
B&B CUT	number of cutting planes added outside root node
TIM	CPU time in seconds (Sun Ultra 250, 2×400 MHz)
GREE	Greedy gap: $(v_{Greedy} - v_0)/v_0 \cdot 100\%$

Table 1 contains information from 30 individual test instances. In these instances the supply graph was complete and contained 70 nodes. The edge costs were generated at random using the uniform distribution in the range $1 - 100$. The demand edges were chosen randomly, in total 35 demand edges. In the initial LP relaxation there are about 5000 variables and constraints. Table 2 contains results based 4000 test runs. Each horizontal line contains the average of the test results for 1000 instances of the specified size. Here NOD and DEM is the number of nodes and demands, respectively. Again the supply graph is complete and the demands are chosen randomly.

v_2	v_1	v_0	GAP1	GAP 2	CUT	NOD	B&B CUT	TIM	GREE
370.0	370.0	370.0	0.0	0.0	6	1	0	1.67	14.59
367.50	371.17	372.00	1.21	0.22	36	3	0	5.45	23.39
368.50	368.50	369.00	0.14	0.14	0	3	0	1.96	14.91
386.00	386.00	386.00	0.00	0.00	0	1	0	1.71	8.81
442.00	446.00	446.00	0.90	0.00	30	1	0	5.15	21.97
333.00	334.00	334.00	0.30	0.00	42	1	0	6.25	29.17
366.00	366.00	366.00	0.00	0.00	42	1	0	1.61	12.57
398.00	398.00	398.00	0.00	0.00	3	1	0	2.29	21.36
328.00	328.00	328.00	0.00	0.00	3	1	0	1.49	5.49
371.00	372.00	372.00	0.27	0.00	95	1	0	9.86	10.48
477.83	479.00	479.00	0.24	0.00	18	1	0	2.86	9.32
403.00	403.00	403.00	0.00	0.00	252	1	0	1.80	11.17
448.00	458.00	458.00	2.18	0.00	122	1	0	7.33	24.45
326.00	326.00	326.00	0.00	0.00	122	1	0	1.60	5.21
351.00	355.00	355.00	1.13	0.00	9	1	0	2.86	8.73
367.50	369.67	370.00	0.68	0.09	30	3	0	3.66	6.22
400.00	401.00	401.00	0.25	0.00	24	1	0	4.55	6.19
407.83	414.00	416.00	1.96	0.48	244	3	0	15.74	17.66
470.00	474.00	474.00	0.84	0.00	213	1	0	16.74	13.29
432.00	432.00	432.00	0.00	0.00	213	1	0	1.76	23.38
408.00	409.50	410.00	0.49	0.12	21	5	0	4.74	24.63
377.00	385.00	385.00	2.08	0.00	51	1	0	7.58	10.39
385.17	389.67	390.00	1.24	0.09	30	3	0	5.56	17.09
350.33	353.00	353.00	0.76	0.00	30	1	0	3.12	16.15
393.00	393.00	393.00	0.00	0.00	30	1	0	1.71	15.27
391.50	393.00	393.00	0.38	0.00	6	1	0	2.58	28.24
466.67	481.00	483.00	3.38	0.41	835	5	0	51.54	19.63
366.00	366.00	366.00	0.00	0.00	835	1	0	1.65	21.31
355.00	359.00	359.00	1.11	0.00	12	1	0	3.31	19.78
366.50	368.00	368.00	0.41	0.00	24	1	0	4.46	7.34

Table 1: Individual result for random demands.

NOD	DEM	GAP 1	GAP 2	CUT	B&B CUT	TIM	GREE
60	30	0.62	0.08	78.3	6.1	5	15.1
80	40	0.48	0.06	85.7	4.2	13	16.4
100	50	0.22	0.02	119.5	0.0	27	17.2
120	60	0.26	0.03	127.8	1.9	74	17.2

Table 2: Random demands, average for 1000 test instances per line.

We also tested the performance of our algorithms on instances with disjoint demand edges. Recall that the problem with this restriction may be solved polynomially using matching techniques. Our interest here was to see if the disjointness of demands had any significant effect on our two algorithms, in particular on the gaps. (In terms of speed a matching algorithm would be better.) The results for 30 individual instances are found in Table 3 while averages over 4000 test runs are in Table 4.

The conclusions of our computational study may be summarized in the following way. We notice that the LP relaxation of the path model seems to give a very good approximation to the underlying integral polytope, see the GAP1 column. Moreover, almost every problem instance is solved to optimality after adding a number of our cutting planes to the initial program. Only a few cases had to be carried out by branch and bound and, in that case, very few branching nodes were required. We also note that the additional cuts strengthens the initial LP relaxation considerably; it typically closes about 90% of the optimality gap of the LP relaxation. It is also interesting to see how the Greedy algorithm performs. It is very fast, usually it takes less than a second on the largest instances. Moreover, the quality of the greedy solution is not too bad, see the GREE column. The gap is about 5-10% for disjoint demands and 15-20% for random demands. We should point out, however, that the lower bounds provided by the greedy algorithm are usually very poor. This means that Greedy fails to recognize that it actually provides quite good primal feasible solutions. The reason for this lies in the fact that the greedy algorithm does not solve the LP dual to optimality. From the table we clearly realize that the greedy algorithm works best for relatively small problem instances, while the cutting plane approach works quite good even for larger problems.

6 Final remarks

We have introduced and studied the 2-path network design problem. Some polynomially solvable subclasses were identified and properties of an LP relaxation was investigated. We have established different classes of cutting planes for a natural model of the problem. Based on extensive numerical testing we conclude that our relaxation, with additional cutting planes, is very tight and that large instances can be solved by our branch-and-cut algorithm. Moreover, our greedy algorithm is very fast and provides solutions that have cost typically about 10% above the optimal value. An interesting topic for further work is to determine other polynomially solvable subclasses of the problem. It would also be interesting to develop fast interchange heuristics to be combined with our greedy algorithm.

v_2	v_1	v_0	GAP1	GAP2	CUT	NOD	B&B CUT	TIM	GREE
416.5	430.0	430.0	3.13	0	173	1	0	17.25	6.51
488.0	489.0	489.0	0.20	0	6	1	0	2.58	2.25
482.0	498.0	498.0	3.21	0	304	1	0	18.10	13.86
379.0	385.0	385.0	1.56	0	94	1	0	8.36	5.97
431.0	431.0	431.0	0	0	0	1	0	1.73	7.42
467.0	467.0	467.0	0	0	0	1	0	1.78	16.49
415.0	422.0	422.0	1.66	0	39	1	0	4.84	4.72
480.5	494.0	494.0	2.73	0	126	1	0	12.87	5.05
384.0	384.0	384.0	0	0	0	1	0	1.73	10.42
444.0	444.0	444.0	0	0	0	1	0	1.77	3.15
361.0	361.0	361.0	0	0	0	1	0	1.78	13.02
456.0	461.1	462.0	1.30	0.19	51	3	0	8.61	4.33
429.5	436.0	436.0	1.49	0.0	123	1	0	18.41	6.42
375.5	379.1	380.0	1.18	0.25	51	3	0	8.54	1.32
437.5	441.0	441.0	0.79	0.0	30	1	0	6.07	7.71
415.0	416.0	416.0	0.24	0.0	18	1	0	4.46	8.41
430.5	435.0	435.0	1.03	0.0	36	1	0	6.96	8.05
431.0	436.0	436.0	1.15	0.0	63	1	0	7.01	9.13
437.5	442.0	442.0	1.02	0.0	30	1	0	6.08	2.71
403.0	403.0	403.0	0	0	12	1	0	3.66	8.93
389.0	397.0	397.0	2.02	0	129	1	0	13.99	10.58
470.5	478.0	478.0	1.57	0	59	1	0	7.59	2.50
471.0	478.0	478.0	1.46	0	75	1	0	8.49	11.30
444.0	444.0	444.0	0	0	0	1	0	1.89	9.91
456.0	466.0	466.0	2.15	0	81	1	0	10.22	15.45
377.0	377.0	377.0	0	0	0	1	0	1.71	3.18
373.0	375.0	375.0	0.5	0	24	1	0	5.33	1.60
430.0	430.0	430.0	0	0	0	1	0	1.92	7.44
423.5	426.0	426.0	0.59	0	81	1	0	12.08	10.05
493.0	493.0	493.0	0	0	0	1	0	1.82	6.90

Table 3: Individual result for disjoint demands.

NOD	DEM	GAP1	GAP2	CUT	B&B CUT	TIM	GREE
60	30	0.39	0.04	55.7	0.32	4	7.8
80	40	0.30	0.03	85.0	0.12	14	7.9
100	50	0.34	0.03	108.5	0.06	33	7.5
120	60	0.31	0.02	127.7	0.09	78	7.5

Table 4: Disjoint demands, average for 1000 test instances per. line.

APPENDIX

Proof of Theorem 4.1.

Let $\mathbf{x}$ be a vertex of LP_{2s} . It follows from Proposition 2.2 that there is a nonnegative “flow vector” $\mathbf{z} \in \mathbb{R}^P$ satisfying $\sum_{P \in \mathcal{P}_d} z_P = 1$ for $d \in D$ and $z_P \leq x_e$ for $e \in P$, $P \in \mathcal{P}_d$, $d \in D$. It is easy to derive that $\mathbf{x} \leq \mathbf{1}$.

The main idea. The proof is based on the following perturbation idea. Assume that we can find distinct demands $d_{i_1}, \dots, d_{i_t}$ and subsets $E_0, E_1, \dots, E_t$ of E such that (i) for $j = 1, \dots, t-1$ the set $E_j := \{e \in E_{i_j, i_{j+1}} : x_e > 0\}$ satisfies $|E_j| \geq 3$, and (ii) E_0 contains a 2-path for demand d_{i_1} and $z_P < x_e$ for each $e \in E_0$ and $P \in \mathcal{P}_d$ with $e \in P$ and $d \neq d_{i_1}$, and (iii) E_t contains a 2-path for demand d_{i_t} and $z_P < x_e$ for each $e \in E_t$ and $P \in \mathcal{P}_d$ with $e \in P$ and $d \neq d_{i_t}$. Then, for each $j = 1, \dots, t$, both E_{j-1} and E_j contain a 2-path for d_{i_j} which may carry positive flow. Consider the sequence

$$(*) \quad E_0, E_1, \dots, E_t$$

and the “signed incidence vector” $\mathbf{y} = \sum_{j=0}^t (-1)^j \chi^{E_j}$. Then, for $\epsilon > 0$ small enough, both the points $\mathbf{x} \pm \epsilon \mathbf{y}$ lie in LP_{2s} . To see this, note first that, for each of these demands d_{i_j} , we may shift a small positive flow from a 2-path in E_{j-1} to a 2-path in E_j . Moreover, as demands are disjoint, and due to properties (ii) and (iii) (concerning E_0 and E_t), other demands than $d_{i_1}, \dots, d_{i_t}$ are unaffected by this perturbation. We call the sequence in $(*)$ a *feasible perturbation* (as it produces new feasible points in LP_{2s}). It follows from the extreme point property that no feasible perturbation can exist (as $\mathbf{x} = (1/2)(\mathbf{x} + \epsilon \mathbf{y}) + (1/2)(\mathbf{x} - \epsilon \mathbf{y})$), and this fact will be used to analyze the structure of the vertex $\mathbf{x}$. Note that there are different slight variations of the perturbation idea. For instance, we may remove E_0 and define $E_t = \{e \in E_{i_t, i_1} : x_e > 0\}$. Then, if t is even, we may define $\mathbf{z}$ as above and (as E_1 and E_t get opposite signs) we have again a feasible perturbation (a “cycle”; see below).

Let D' be the set of demands with “fractional flow”, i.e., $d \in D'$ means that $z_P < 1$ for all $P \in \mathcal{P}_d$. Define a graph Γ with node set D' and an edge $g_{i,j} = [d_i, d_j]$ between d_i and d_j whenever $|E(g_{i,j})| \geq 3$ where $E(g_{i,j}) := E_{i,j} \cap \{e \in E : x_e > 0\}$.

Claim 1: Γ has no even cycle.

Proof of Claim 1: Assume that Γ has an even cycle with edges $g_1, g_2, \dots, g_t$. Then it is easy to verify that

$$E(g_1), E(g_2), \dots, E(g_t)$$

is a feasible perturbation; a contradiction. Thus Claim 1 holds.

Claim 2: Each component of Γ is a tree or an odd dumbbell.

Proof of Claim 2: Consider a nontree component K in Γ . Thus, K contains a cycle C_Γ which must be odd (due to Claim 1). It suffices to prove that C_Γ must be the unique cycle in K . Let C'_Γ be another cycle in K . Then $C_\Gamma \cap C'_\Gamma$ contains at most one node; otherwise it is easy to see that $C_\Gamma \cup C'_\Gamma$ would contain an even cycle. Thus, C_Γ and C'_Γ are edge-disjoint odd cycles connected by some

path P_Γ . But then one can find a feasible perturbation involving the sets $E(g)$ for $g \in C_\Gamma \cup P_\Gamma \cup C'_\Gamma$ (use alternating signs on consecutive edges such that every node in $C_\Gamma \cup P_\Gamma \cup C'_\Gamma$ meets edges with both signs). This proves Claim 2.

We may now strengthen the previous claim.

Claim 3: Each component of Γ is either a single edge or an odd cycle.

Proof of Claim 3: Due to Claim 2 we need to prove that (1.) each odd dumbbell component if Γ is just a cycle, and (2.) each tree component is just an edge.

(1): Assume that an odd dumbbell component is not a cycle. Then there is a path P_Γ from a node in the (unique) cycle C_Γ to some leaf d_{i_1} of the dumbbell component. P_Γ consists of the edges $[d_{i_1}, d_{i_{i+1}}]$ for $i = 1, \dots, t$ where d_{i_t} is the endnode of P_Γ that lies in C_Γ . We want to find a feasible perturbation. To do so, we initially consider the sets $E(g)$ for each edge g lying in C_Γ or P_Γ except the last edge $[d_{i_2}, d_{i_1}]$. Let $\mathcal{S}$ denote the sequence of these sets. We associate alternate signs to these sets as in the proof of Claim 2 (alternating signs on consecutive edges such that every node meets edges with both signs). The only problem is how to finish at d_{i_2} and d_{i_1} to get a feasible perturbation.

First, assume that there is a 2-path P (in G) for d_{i_1} such that P does not contain any of the endnode of demand edge d_{i_2} . Then we may append P to the mentioned edge sets $E(g)$ and thereby get a feasible perturbation. The reason is that only demand d_{i_1} has flow on edges in P . Next, we consider the alternative situation where no such path P exists. Then, all flow for demand d_{i_1} is on the two paths P_1 and P_2 that uses edges in E_{i_1, i_2} . Moreover, both paths carry positive flow as $d_{i_1} \in D'$. We discuss two possible cases.

Case 1: $x_e \geq 1/2$ for each $e \in E_{i_1, i_2}$. Then we let the perturbation sequence simply be $\mathcal{S}$ which is feasible as demand d_{i_2} may be satisfied by sending a flow of $1/2$ on each of the two 2-paths for d_{i_2} in E_{i_1, i_2} .

Case 2: $x_e < 1/2$ for at least one edge e in E_{i_1, i_2} . We may assume that P_1 has flow α and P_2 flow $1 - \alpha$ where $\alpha > 1/2$. Thus, P_2 has at least one edge e with $x_e < 1/2$ while the edges in P_1 both have x -value larger than $1/2$ (in fact, at least α). Then we let the perturbation sequence be

$$\mathcal{S}, P_2, P_1.$$

This is a feasible perturbation. In fact, for demand d_{i_1} flow is just shifted between its two 2-paths P_1 and P_2 and for demand d_{i_2} , when flow is reduced towards d_{i_3} there is an increase of flow towards d_{i_1} (as the capacities of the paths P_1 and P_2 get closer to each other).

(2): Consider a tree component with at least three nodes. Then it contains two leaves d_{i_1} and d_{i_t} and we let P_Γ denote the (unique) path connecting these two nodes. One may find a feasible perturbation as we did in the previous paragraph. The only difference is that now both endnodes d_{i_1} and d_{i_t} of P_Γ are treated as we did for the single endnode of the path P_Γ in the dumbbell case (i). It follows that each tree component of Γ must be a single edge. (It is easy to find a feasible perturbation for a component which is a single node.)

This proves Claim 3.

We now observe that we may strengthen Claim 3 as follows. Due to Claim 3 each node d in Γ is adjacent to one node (when the component is an edge) or two nodes (when the component is an odd cycle). We call node d *free* if there is a 2-path P for d (in G) such that (i) $x_e > 0$ for each $e \in P$, and (ii) P does not intersect any of the demands that are adjacent to d (i.e., if d is adjacent to d' and d'' then no node in P is an endnode of d' or d''). With a slight variation in the arguments used to prove Claim 3 we now obtain the following main result.

Claim 4: Each component of Γ is either a single edge or an odd cycle. Moreover, no node in Γ is free.

We are now, finally, able to actually determine the numerical values on the components of $\mathbf{x}$!

Consider first a component of Γ which is a single edge, say (for simplicity) $[d_1, d_2]$. Since neither $d_1, d_2 \in D'$ and no node is free, we must have that $x_e > 0$ for each $e \in E_{1,2}$. Assume that $x_e < 1/2$ for an edge $e \in E_{1,2}$. Then $x_f > 1/2$ for each of the other three edges in $E_{1,2}$ (otherwise the flow for one of the demands would be less than one). But then we may perturb $\mathbf{x}$ by letting $x_e := x_e + \epsilon$ and $x_f := x_f - \epsilon$ for $f \in E_{1,2} \setminus \{e\}$ (and vice versa) and thereby contradict the extreme point property. This proves that $x_e \geq 1/2$ for each $e \in E_{1,2}$ and therefore (again by the extreme point property) we must have $x_e = 1/2$ for each $e \in E_{1,2}$.

Consider next a component of Γ which is an odd cycle C_Γ (with only free nodes). Assume that $[d_1, d_2]$ is an edge in the cycle, and assume that $\{e \in E_{1,2} : x_e > 0\}$ consists of the three edges e, f and g . We may assume that $\{e, f\}$ is the 2-path for demand d_2 and that f also lies in a 2-path for d_1 . Note that e is not in a 2-path for d_1 and therefore, using the extreme point property, we deduce that $x_e \leq x_f$. Assume that $x_e < x_f$. Then we find a feasible perturbation as follows: start with sets $E(g)$ for each $g \in C_\Gamma$ and give signs such that the sets associated with the two edges incident to d_2 gets the sign $-$. Then, modify this set sequence by replacing $E(g)$, where $g = [d_1, d_2]$, by the two sets $\{f, g\}$ and $\{e\}$ with signs $-$ and $+$, respectively. The fact that this is a feasible perturbation implies that we must have $x_e = x_f$, and by symmetry, $x_e = x_f = x_g$. Similar techniques may be used to prove that there are numbers α_g for $g \in C_\Gamma$ such that $x_e = \alpha_g$ for each $e \in E(g)$, $g \in C_\Gamma$. And, finally one may derive that α_g is $1/2$ or $1/4$ according to the cardinality of the set $\{e \in E_{i,j} : x_e > 0\}$ associated with g .

This proves the desired form of all vertices of LP_{2s} . All these points are actually vertices; the proof of this fact is fairly straightforward and omitted.

References

- Dahl, G. (1998), ‘The 2-hop spanning tree problem’, *Operations Research Letters* **23**, 21–26.

- Dahl, G. (1999), ‘Stable set polytopes for a class of circulant graphs’, *SIAM Journal of Optimization* **9**(2), 493–503.
- Goemans, M. & Williamson, D. (1995), ‘A general approximation technique for constrained forest problems’, *SIAM J. Computing* **24**(2), 296–317.
- Gouveia, L. (1996), Using variable redefinition for computing minimum spanning and steiner trees with hop constraints, Technical Report 2, Faculdade de Ciências da Universidade de Lisboa, Centro de investigação operacional, Lisboa, Portugal.
- Grötschel, M., Lovász, L. & Schrijver, A. (1988), *Geometric algorithms and combinatorial optimization*, Vol. 2 of *Algorithms and combinatorics*, Springer.
- Li, C., McCormick, S. T. & Simchi-Levi, D. (1992), ‘On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems’, *Operations Research Letters* **11**, 303–308.
- Nemhauser, G. & Wolsey, L. (1988), *Integer and combinatorial optimization*, Wiley.
- Schrijver, A. (1986), *Theory of linear and integer programming*, Wiley, Chichester.