Available online at www.sciencedirect.com

ScienceDirect

Discrete Optimization 5 (2008) 314–322

DISCRETE OPTIMIZATION

www.elsevier.com/locate/disopt

The k-path tree matroid and its applications to survivable network design

Esther M. Arkin[a,*], Refael Hassin[b]

[a] *Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794-3600, United States*
[b] *School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel*

Received 16 November 2005; accepted 12 November 2006
Available online 5 November 2007

Abstract

We define the k-path tree matroid, and use it to solve network design problems in which the required connectivity is arbitrary for a given pair of nodes, and 1 for the other pairs. We solve the problems for undirected and directed graphs. We then use these exact algorithms to give improved approximation algorithms for problems in which the weights satisfy the triangle inequality and the connectivity requirement is either 2 among at most five nodes and 1 for the other nodes, or it is 3 among a set of three nodes and 1 for all other nodes.
© 2007 Elsevier B.V. All rights reserved.

MSC: 68R99; 05C40

Keywords: Network connectivity; Algorithm; Matroid intersection

1. Introduction

Consider a complete graph $G = (V, E)$ with nonnegative edge weights w_{ij} for each $(i, j) \in E$. Denote the weight of a subgraph G' of G by $w(G') = \sum_{(i,j)\in G'} w_{ij}$. THE SURVIVABLE NETWORK DESIGN PROBLEM (SND), seeks a minimum weight subgraph of G such that each pair of nodes i, j has a pre-specified requirement r_{ij} of edge-disjoint $i - j$ paths. When $r_{ij} = 1$ for all i, j, this is the classical MINIMUM SPANNING TREE PROBLEM. When $r_{ij} \in \{0, 1\}$, this is the well-known NP-hard MINIMUM STEINER TREE PROBLEM. Jain [14] presented a 2-approximation algorithm for the SND problem doing integer rounding to a linear programming relaxation.

The k-PATH TREE PROBLEM is the special case of the SND problem in which there are two special nodes p and q such that $r_{pq} = k$ and $r_{ij} = 1$ for all $\{i, j\} \neq \{p, q\}$.

Balakrishnan, Magnanti, and Mirchandani [2] considered the 2-PATH TREE PROBLEM. They show that if the edge weights satisfy the triangle inequality, then the problem can be solved in polynomial time using matroid intersection algorithms.

* Corresponding author.
E-mail addresses: estie@ams.sunysb.edu (E.M. Arkin), hassin@post.tau.ac.il (R. Hassin).
URLs: http://www.ams.sunysb.edu/~estie/ (E.M. Arkin), http://www.math.tau.ac.il/~hassin/ (R. Hassin).

1572-5286/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2006.11.009

The same authors [3] provide a 2-approximation algorithm for the k-PATH STEINER TREE PROBLEM which is a generalization of the k-PATH TREE PROBLEM, where $r_{pq} = k$ for the given pair p, q, and $r_{ij} \in \{0, 1\}$ otherwise, assuming that the weights w_{ij} satisfy the triangle inequality. In [4], the same authors consider the following SND problem: Each node i has a nonnegative integer ρ_i, $r_{ij} = \min\{\rho_i, \rho_j\}$ for all $(i, j) \in E$. They propose the following algorithm: In the first step, approximate the corresponding *backbone network problem* which consists of meeting the requirements of all $r_{ij} \geq 2$ values. In the second step, complete the *access network* to meet the remaining requirements. This step is accomplished by computing a 2-approximation to a Steiner forest problem on a contracted graph. The performance guarantee of this algorithm is $1 + \theta$ where θ is the approximation factor of the first step.

In this paper we define the k-path tree matroid. We then use this matroid to extend the results of [2] in several ways: In Section 3 we give a polynomial time algorithm for the k-PATH TREE PROBLEM. In Section 4 we use this exact algorithm to give improved approximation algorithms for some problems assuming that the weights satisfy the triangle inequality. In Section 5 we consider the problem on directed graphs, where r_{ij} denotes the requirement of edge-disjoint directed paths from node i to node j. We give a polynomial time algorithm when $r_{1q} = 2$ for specified nodes 1 and q, and $r_{1j} = 1$ for all other nodes $j \in V$. We prove that simple extensions of this problem are already hard.

The SND problem was first introduced by Steiglitz, Weigner, and Kleitman [23]. The survey by Grötschel, Monma, and Stoer [11] describes motivation, polynomially solvable cases, Integer Programming methods, heuristics, and directed variants. See also [13] for a recent survey. Frank [8] considers the directed SND problem where for every subset of nodes $S \subseteq V$ the number of arcs leaving S is at least $f(S)$. He shows that for requirement function $f(S)$ that is intersecting supermodular, the problem is polynomial. Melkonian and Tardos [18] give a 2-approximation when f is crossing supermodular.

In this paper disjoint paths mean edge-disjoint paths, unless stated otherwise. We denote by *opt* the value of an optimal solution to the problem under consideration.

2. The *k*-path tree matroid

Consider a complete graph $G = (V, E)$ with a distinguished node p and n nodes in total. Let $k \leq n - 1$ be a constant. Define the bases of E as subsets of E with exactly $n + k - 2$ edges which induce a spanning subgraph, all of whose cycles contain p. Let $\mathcal{F}$ denote the family of subsets of the above bases. Consider the *k-path tree matroid* $\mathcal{M}_{p,k} = (E, \mathcal{F})$ to be the system defined by these bases. Below we prove that this is a matroid.

$\mathcal{M}_{p,1}$ is the graphic matroid. $\mathcal{M}_{p,2}$ is sometimes called the 1-tree matroid ([1] Exercise 13.38).[1] We prove that it is a matroid also for $k > 0$.

Theorem 2.1. *$\mathcal{M}_{p,k} = (E, \mathcal{F})$ is a matroid.*

Proof. We use the following well-known construction: The *matroid sum* of a matroid $\mathcal{M}_1$ on set E_1 and a matroid $\mathcal{M}_2$ on set E_2 is the matroid $\mathcal{M}$ on $E_1 \cup E_2$ where each independent set is the union of an independent set of $\mathcal{M}_1$ and an independent set of $\mathcal{M}_2$ ([19], see also [15]).

Let E be the edge set of the graph G, and define E' to be the set of edges incident at node p. Let $\mathcal{M}$ denote the *uniform matroid* on E', where each set of cardinality $\leq k - 1$ is independent.

We first claim that each base of $\mathcal{M}_{p,k}$ is a spanning tree of G plus $k - 1$ edges incident at p. To see this property, consider a base B. While B contains more than $n - 1$ edges, it contains a cycle. By assumption, the cycle contains two edges incident with p. Remove one of these two edges. After $k - 1$ repetitions of this process we end with a spanning tree, proving the claim. Therefore, $\mathcal{M}_{p,k}$ is the matroid sum of the uniform matroid on E' with rank $k - 1$ and of the graphic matroid of G. ■

3. The *k*-path tree problem

The k-PATH TREE PROBLEM with special nodes $p, q \in V$ requires to compute a minimum weight spanning subgraph that contains k disjoint $p - q$ paths, and 1 path between any other pair of nodes.

[1] The bases of this matroid are different from the 1-trees defined in [12] since the degree of the special node is not restricted to be 2. See also, [10,24].

Theorem 3.1. *Consider the k-PATH TREE PROBLEM with special nodes $p, q \in V$, and add to it the requirement that the k paths are node-disjoint. Then, the problem is equivalent to finding a minimum weight common base of the matroids $\mathcal{M}_{p,k}$ and $\mathcal{M}_{q,k}$.*

Proof. Clearly, a solution to the k-PATH TREE PROBLEM contains a common base to these matroids. We now show the other direction. Consider a common base B. It induces a connected subgraph because this property is required from every base of either matroid. Moreover, every cycle in B includes both p and q. Consider the subgraph $G' = (V', E')$ induced by the union of all the cycles of B, and suppose that it contains n_0 nodes. Since every node in $V \setminus V'$ needs a single edge to maintain connectivity, $|E'| = |B| - (n - n_0) = (n + k - 2) - (n - n_0) = n_0 + k - 2$. Consider a cycle induced by B, i.e., a pair of $p - q$ paths. These paths must be node-disjoint since otherwise they contain a cycle that does not include one of the nodes p and q. The following argument proves that G' contains k node-disjoint $p - q$ paths: Initially we have $|E'| - |V'| = k - 2$. While $|E'| - |V'| \geq 0 G'$ contains a cycle consisting of two node-disjoint $p - q$ paths. Choose one of these paths and delete its edges and internal nodes. This change decreases $|E'| - |V'|$ by 1. After $k - 1$ repetitions we get $|E'| - |V'| = -1$ and hence a spanning tree, containing exactly one $p - q$ path. Altogether we have identified k node-disjoint $p - q$ paths. ■

Corollary 3.2. *The k-PATH TREE PROBLEM with special nodes $p, q \in V$, and weights that satisfy the triangle inequality is equivalent to finding a minimum weight common base of the matroids $\mathcal{M}_{p,k}$ and $\mathcal{M}_{q,k}$.*

Proof. We observe that under the triangle inequality any solution can be converted, without increasing its weight, to one where the k $p - q$ paths are node-disjoint. (While there exist two paths with a common node, shortcut one of these paths so it skips this node.) By the triangle inequality, this change does not increase the cost of the solution. Therefore Theorem 3.1 applies. ■

Theorem 3.3. *The k-PATH TREE PROBLEM can be solved in polynomial time.*

Proof. We want to formulate the problem as a matroid intersection problem. However, when the triangle inequality does not hold, the edge-disjoint $p - q$ paths in an optimal solution are not necessarily node-disjoint, and therefore we do not know the number of edges used by the solution. We overcome this difficulty by transforming the graph as follows. For each node $u \in V$ denote by $\delta(u)$ the set of edges that are incident with u in G. Replace $u \neq p, q$ by a set of nodes $\{u_e : e \in \delta(u)\}$. Each edge $(u, v) \in E$ is replaced by an edge (u_e, v_e) with the same weight, where $u_e \equiv p$ for $u = p$ and $u_e \equiv q$ for $u = q$. We also add zero-weight edges (u_e, u_f) for every $u_e, u_f \in \delta(u)$. We now solve the matroid intersection problem as in the proof of Theorem 3.1.

The problem of finding a minimum weight common base of two matroids can be solved in polynomial time (see, for example, [22]). ■

4. Approximation algorithms

In this section we apply Theorem 3.3 and present several approximation algorithms with bounded performance guarantees for SND problems in which more than two nodes require connectivity larger than 1, and *the weights satisfy the triangle inequality*. Since Jain's algorithm [14] can be used to obtain a 2-approximation for the general SND problem, we are only interested in cases where a better approximation can be achieved. In particular, for a problem with a set S of nodes, $|S| \leq 5$, and requirement $r_{ij} = 2$ for every $i, j \in S$ and $r_{ij} = 1$ otherwise, we present a $\left(2 - \frac{2}{|S|}\right)$-approximation algorithm. We also present a $\frac{11}{7}$-approximation when $|S| = 3$, $r_{ij} = 3$ for every $i, j \in S$ and $r_{ij} = 1$ otherwise.

We denote by OPT the solution of the problem under consideration, and by *opt* the cost (value) of this solution. Recall that throughout this section we assume that G is a complete graph and that the edge weights satisfy the triangle inequality.

Lemma 4.1. *Consider an SND problem with $r_{ij} = 2$ for all $i, j \in S$, where $|S| = s \leq 5$. Let G_S be the subgraph of G induced by S. Let H be a minimum cost Hamiltonian cycle in G_S. Then, $w(H) \leq opt$.*

Proof. It suffices to consider the case $s = 5$, since for $s < 5$ we can duplicate nodes in S, with edge costs zero between duplicate nodes. This does not change the cost of an optimal solution, nor the minimum cost of a Hamiltonian cycle.

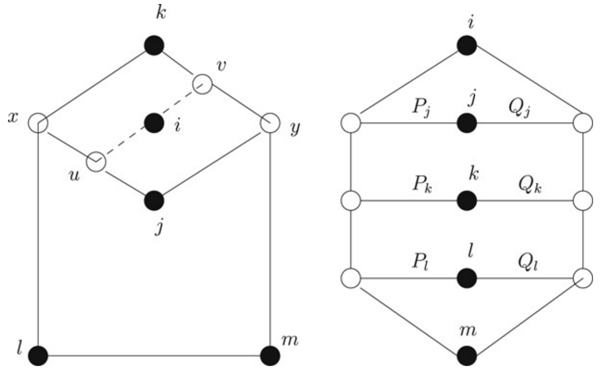

Fig. 1. Subgraphs defined by S: Case 3 (left) and 4 (right).

We prove the lemma by considering a cycle C contained in OPT containing as many nodes of S as possible.
Case 1: $|C \cap S| = 5$. In this case we can simply shortcut C into a cycle containing only the nodes of S, by the triangle inequality, and obtain the desired Hamiltonian cycle.
Cases 2–4: $|C \cap S| < 5$. In these cases, we create two closed walks, CW_1 and CW_2 each of which contains all five nodes of S, and their edges are contained in OPT. In our construction, each edge of OPT appears at most twice in the edges of $CW_1 \cup CW_2$, and so we have $w(CW_1) + w(CW_2) \leq 2opt$. Without loss of generality, assume that $w(CW_1) \leq w(CW_2)$ so we have $w(CW_1) \leq opt$. By shortcutting this closed walk we get the desired Hamiltonian cycle.
Case 2: $|C \cap S| = 4$. Let node $k \in S \setminus C$. Consider the two paths from k to one of the other nodes $i \in S \cap C$, and denote by P_k, Q_k the parts of the paths until they hit the cycle. Now create a closed walk CW_1 visiting the five nodes, contained in OPT, by concatenating the cycle with two copies of P_k. Similarly, create a second closed walk CW_2 visiting all the five nodes of S, contained in OPT, by concatenating the cycle with two copies of Q_k.
Case 3: $|C \cap S| = 3$. We first claim that there must be a cycle $\hat{C}$ containing three nodes from S, and four disjoint paths P_1, P_2, Q_1, Q_2 such that P_1 and Q_1 connect a fourth node of S to $\hat{C}$, and P_2 and Q_2 connect the fifth node from S to $\hat{C}$.

Consider a cycle C with three nodes, say k, l, m from S, and the subgraph induced by C and two paths connecting j to C. This subgraph is shown by the solid lines in Fig. 1 (left). Note that this is the only way that has no cycle with all of these four nodes. Let C' be the cycle defined by (x, j, y, m, l, x). Now consider two paths connecting i to this subgraph. These paths are shown in the figure by dashed lines. For any other pattern of these path either there a is cycle with four nodes of S, or one of C and C' satisfies the requirements of our claim. However, now the cycle defined by (x, u, i, v, y, m, l, x) with the paths connecting j and k to it satisfy the requirements.

Consider now the cycle C and paths that satisfy the claim. For example, suppose that i, j, k are on C, and denote by P_l and Q_l the paths connecting l to C, and similarly P_m and Q_m connect m to C. Recall that the four paths are disjoint. We construct one walk from C, two copies of P_l and two copies of P_m, and the second walk has C, and two copies of Q_l and Q_m.
Case 4: $|C \cap S| = 2$. The situation is shown in Fig. 1 (right). Let C be the cycle consisting of the paths connecting i and m. One walk consists of C and two copies of P_j, P_k and P_l. The other walk consists of C and two copies of Q_j, Q_k and Q_l. ■

It is interesting to note that the lemma cannot be extended to $|S| = 6$ as the following example shows: Let G be a graph that has 8 nodes, a, b, c, d, e, f, g, h, with $S = \{c, d, e, f, g, h\}$, $|S| = 6$. The following edges have cost 1: $(a, c)(a, e)(a, g)(c, d)(e, f)(g, h)(d, b)(f, b)(h, b)$. Other edges have costs induced by this metric. For example (c, e) has cost 2. A feasible solution to the SND has cost 9, using the 9 edges of cost 1. However, every cycle in G that contains the nodes of S has cost larger than 9.

Theorem 4.2. *Let $|S| = s$, where $s \leq 5$. The SND problem with $r_{ij} = 2$ for $i, j \in S$, and $r_{pq} = 1$ otherwise, can be approximated with a factor of $2 - \frac{2}{s}$.*

Proof. Denote by H a minimum weight Hamiltonian cycle in the subgraph induced by S. By Lemma 4.1, $w(H) \leq opt$. Suppose that (i, j) has maximum weight over the edges of H, and let $e = (k, l)$ be the next heaviest edge in H.

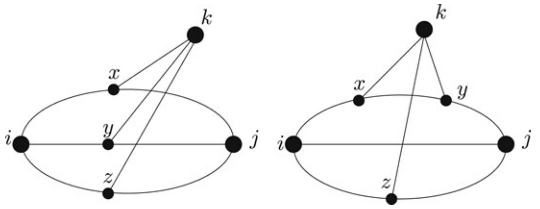

Fig. 2. The subgraph G' in the proof of Lemma 4.3.

W.l.o.g. assume that (i, k, l, j) appear in H in this order. Let T be an optimal solution to the 2-path tree problem with special nodes i and j. Let C be the unique cycle of T. Call $v \in S$ a *leaf* in T if $v \in \{i, j, k, l\}$ or $v \notin C$ and there is no $u \in S \setminus C$ such that v is on the unique path in T connecting u and C. We observe that to form a feasible solution, it is sufficient to add to T edges so that the set of leaves is 2-connected. Let H' be the cycle obtained from H by shortcutting nodes that are not leaves. By the triangle inequality, $w(H') \le w(H)$. Let MG be the multigraph obtained by the edges of T with the edges of $H' \setminus \{(i, j)\}$. We observe that MG satisfies the connectivity requirements, namely it is connected and has two disjoint paths between any pair of nodes from S.

We claim that also $MG \setminus \{e\}$ satisfies the connectivity requirements. To prove this property it is sufficient to observe that each of the nodes k and l which is not in C has two disjoint paths to C in $MG \setminus \{e\}$. One path uses T and the other one uses $H' \setminus \{(i, j), e\}$. Since (i, j) and e are the heaviest edges in H', $w(MG \setminus \{e\}) \le \left(2 - \frac{2}{s}\right) opt$.

Note that the only edges that may have two copies in $MG \setminus \{e\}$ are (i, k) and (j, l) (there are no other edges in T between leaves, by definition).

Suppose that $(i, k) \in H' \cap T$. If $k \in C$ then we can simply delete one copy of (i, k) without hurting connectivity. We therefore assume that $k \notin C$, and let $(i, q) \in C$, $q \neq j$. We replace in $MG \setminus \{e\}$ one copy of (i, k) and the edge (i, q) by the edge (k, q). By the triangle inequality, $w_{kq} \le w_{ik} + w_{iq}$.

Similarly, if $(j, l) \in H' \cap T$. The result is a graph that satisfies the connectivity requirements and its weight is at most $w(MG \setminus \{e\}) \le \left(2 - \frac{2}{s}\right) opt$.

Since we assumed $s \le 5$, the algorithm is polynomial. ■

We observe that the two paths connecting every pair of nodes in S generated by our approximation algorithm are node disjoint (in addition to being edge-disjoint). The proofs of Lemma 4.1 and Theorem 4.2 also hold for the node-disjoint case. Thus our approximation algorithm also give an approximation for the problem requiring node-disjoint paths.

The following example demonstrates that the bound of Theorem 4.2 is tight. Let $V = S \cup S'$, where $S = \{v_1, \ldots v_s\}$ and $S' = \{v'_1, \ldots v'_s\}$. All edge weights are 1 except for $w_{v_i v'_i} = 0$, $i = 1, \ldots s$. Clearly, an optimal solution consists of any Hamiltonian cycle of S and the zero-weight edges, so that $opt = s$. Let H be the cycle $(v_1, v_2, \ldots, v_s, v_1)$, and choose (v_1, v_2) for its heaviest edge. Let T consist of (v_1, v_2), the path $v'_1, \ldots, v'_s$ and the zero-weight edges. Then, T solves the 2-path tree problem with special nodes v_1 and v_2, and $w(T) = s$. The approximation SOL is obtained by adding to T $s - 2$ edges from $H \setminus (v_1, v_2)$, and $w(\mathrm{SOL}) = 2s - 2$.

We now consider a problem with connectivity requirement $r_{uv} = 3$ for all $u, v \in S$ and $r_{uv} = 1$ otherwise.

Lemma 4.3. *Consider an SND problem with $r_{ij} = 3$ for all $i, j \in S$. Let $\{i, j, k\} \subseteq S$, then $w_{ij} + w_{ik} + w_{jk} \le \frac{6}{7} opt$.*

Proof. Consider a subgraph G' that contains three disjoint $i - j$ paths P_1, P_2, and P_3. We consider two cases (see Fig. 2, and note that some of the subpaths illustrated in the figure may be degenerate). 1. There are nodes $x \in P_1$, $y \in P_2$, and $z \in P_3$ with a $k - x$ path P_{kx}, a $k - y$ path P_{ky}, and a $k - z$ path P_{kz} disjoint among themselves and disjoint from the three $i - j$ paths. 2. As in (1) except that $x, y \in P_1$, so that $P_1 = P_{ix} \cup P_{xy} \cup P_{yj}$.

Denote by P_{ik} the subpath of P_1 between the specified nodes. Similarly for other subpaths.

Case 1. By the triangle inequality we have:

$$[w(P_{ix}) + w(P_{xj})] + [w(P_{iy}) + w(P_{yj})] + [w(P_{iz}) + w(P_{zj})] \ge 3w_{ij},$$
$$[w(P_{ix}) + w(P_{xk})] + [w(P_{iy}) + w(P_{yk})] + [w(P_{iz}) + w(P_{zk})] \ge 3w_{ik},$$
$$[w(P_{jx}) + w(P_{xk})] + [w(P_{jy}) + w(P_{yk})] + [w(P_{jz}) + w(P_{zk})] \ge 3w_{jk}.$$

Summing and dividing by 2 and noting that the left-hand side is bounded by the weight of G', we get $w(G') \geq \frac{3}{2}(w_{ik} + w_{kj} + w_{ij})$.
Case 2. By the triangle inequality we have:

$$[w(P_{ix}) + w(P_{xy}) + w(P_{yj})] + w(P_2) + [w(P_{iz}) + w(P_{zj})] \geq 3w_{ij},$$
$$[w(P_{ix}) + w(P_{xk})] + [w(P_2) + w(P_{jy}) + w(P_{yk})] + [w(P_{iz}) + w(P_{zk})] \geq 3w_{ik},$$
$$[w(P_2) + w(P_{ix}) + w(P_{xk})] + [w(P_{jy}) + w(P_{yk})] + [w(P_{jz}) + w(P_{zk})] \geq 3w_{jk},$$
$$w(P_3) \geq w_{ij},$$
$$w(P_{iz}) + w(P_{zk}) \geq w_{ik},$$
$$w(P_{jz}) + w(P_{zk}) \geq w_{jk}.$$

We divide the first three inequalities by 3, and the last three inequalities by 6, and add them up. Note that the left-hand side is bounded by the weight of G'. We get $w(G') \geq \frac{7}{6}(w_{ik} + w_{kj} + w_{ij})$. ■

Theorem 4.4. *Let $|S| = 3$. The SND problem with $r_{ij} = 3$ for $i, j \in S$, and $r_{ij} = 1$ otherwise, can be approximated with a factor of $\frac{11}{7}$.*

Proof. Let $S = \{i, j, k\}$. Without loss of generality, $w_{ij} \geq w_{ik}, w_{jk}$. By Lemma 4.3, $w_{ik} + w_{jk} \leq \frac{2}{3} \cdot \frac{6}{7} opt = \frac{4}{7} opt$.

Let OPT be an optimal solution, and consider SOL $=$ OPT $\setminus \{(i, k), (j, k)\}$. We claim that SOL is connected and has two disjoint $i - j$ paths. The claim is obvious if at most one of the two edges is in OPT. If $(i, k), (j, k) \in$ OPT then deleting them removes from OPT one $i - j$ path and the $i - j$ connectivity decreases by 1. The $k - i$ connectivity and $k - j$ connectivity decrease by at most 2. We conclude that the weight of an optimal solution $T'_{i,j}$ to the 2-path tree problem with special nodes i and j in the graph $G \setminus \{(i, k), (k, j)\}$, satisfies $w(T'_{i,j}) \leq opt$. We compute such a solution and call it $T'_{i,j}$. Define $T' = T'_{i,j} \cup \{(i, k), (k, j)\}$, then T' is a feasible solution to our problem, of weight at most $\frac{11}{7} opt$. ■

5. Directed *k*-path tree problems

In this section we consider directed graphs. In our terminology, an arc is a directed edge and disjoint paths are arc-disjoint (allowing the use of an arc by one path and its oppositely directed arc by another path).

Consider a complete directed arc-weighted graph $D = (V, A)$ with nonnegative arc weights w_{ij} for each $(i, j) \in A$. THE DIRECTED SURVIVABLE NETWORK DESIGN PROBLEM (directed SND) seeks a minimum weight subgraph of G such that each pair of nodes i, j has a pre-specified requirement r_{ij} of arc-disjoint $i - j$ directed paths.

The DIRECTED k-PATH TREE PROBLEM with a special source node 1 and a destination node q requires to compute a minimum weight subgraph that contains k arc-disjoint directed paths from 1 to q and one directed path from 1 to every other node.

Let $\delta^{in}(u)$ be the set of arcs entering $u \in V$. Clearly, these sets partition A. We define upper bounds k_u $u \in V$ as follows: $k_1 = 0$, $k_q = k$, and $k_u = 1$ for $u \neq 1, q$. With the sets $\delta^{in}(u)$ $u \in V$ these bounds define a *partition matroid* $\mathcal{M}'_{1,q}$.

Theorem 5.1. *Consider the* DIRECTED k-PATH TREE PROBLEM *with special nodes $1, q \in V$, and add to it the requirement that the k $1 - q$ paths are node-disjoint. Then, the problem can be solved in polynomial time.*

Proof. We claim that the problem is equivalent to finding a minimum weight common base of the two matroids $\mathcal{M}_{1,k}$ (defined on the underlying undirected graph) and $\mathcal{M}'_{1,q}$. A base in the intersection contains a spanning tree T and $k - 1$ additional arcs that enter q and create cycles in the undirected underlying graph, all of which include node 1. In particular, the additional arcs leave nodes $j_1, \ldots, j_{k-1}$ such that the $1 - j_i$ paths in T for $i = 1, \ldots, k - 1$ and the $1 - q$ path in T are all node-disjoint (otherwise, their underlying undirected graph has a cycle that does not contain node 1, a contradiction to the definition of a base in $\mathcal{M}_{1,k}$). Therefore, these paths are node-disjoint $1 - q$ paths. ■

Corollary 5.2. *The* DIRECTED k-PATH TREE PROBLEM *with special nodes $1, q \in V$, and weights that satisfy the triangle inequality is equivalent to finding a minimum weight common base of the matroids $\mathcal{M}_{1,k}$ and $\mathcal{M}'_{1,q}$.*

Proof. The proof is similar to that of Corollary 3.2. ■

Theorem 5.3. *The* DIRECTED k-PATH TREE PROBLEM *can be solved in polynomial time.*

Proof. We use a similar transformation of the graph to the one in the proof of Theorem 3.3. For each node $u \in V$ denote by $\delta(u)$ the set of arcs that are incident with u in G. Replace $u \neq 1, q$ by a set of nodes $\{u_e : e \in \delta(u)\}$. Each arc $(u, v) \in A$ is replaced by an arc (u_e, v_e) with the same weight, where $v_e \equiv q$ for $v = q$ and $u_e \equiv 1$ for $u = 1$. We also add zero-weight arcs (u_e, u_f) for every $u_e, u_f \in \delta(u)$ such that e enters u and f is an arc that leaves u. We now solve the matroid intersection problem as in the proof of Theorem 5.1. ■

We note that for the matroid intersection problem in which one of the matroids is a partition matroid there are specialized algorithms given in [5,9].

We now consider two simple variations of the above problem, and show that (surprisingly) they are NP-complete.

Theorem 5.4. *Given a directed graph with n nodes, and special nodes p and q. Deciding whether there is a solution to the SND problem with $r_{pj} = 1$ for all nodes j, and $r_{qp} = 1$, containing at most n arcs is NP-complete.*

Proof. We use a reduction from the following directed subgraph homeomorphism problem which was shown to be NP-complete in [7]: Does a given directed graph D with three distinguished nodes u, v, and w have a simple $u - w$ path which contains v?

Given an instance of this problem we remove all nodes which are not reachable from u, and all arcs into u. We also add an arc (w, u) to obtain a directed graph D'. Let D' have n nodes. These changes obviously do not affect the existence of the desired path. We define an instance of SND with $r_{uj} = 1$ for all $j \neq u$ and $r_{vu} = 1$. (In other words $u = p$ and $v = q$.) We claim that a $u - w$ simple path containing v exists in D if and only if the SND problem has a solution containing n arcs.

If the SND has a solution, it must use the arc (w, u) since this is the only arc that enters u, and this is necessary to satisfy the requirement $r_{vu} = 1$. If it has a solution with n arcs then it is an arborescences rooted at u and the arc (w, u). Moreover, it is necessary that w is a descendant of v in this arborescence. This yields the desired path in D.

Conversely, if such a path exists in D then it also exists in D'. It is easy to add arcs to form an arborescence. This meets the requirements $r_{uj} = 1$ for all $j \neq u$ in D'. Add also the arc (w, u) to meet the requirement $r_{vu} = 1$. By definition, this solution has n arcs. ■

Theorem 5.5. *The SND problem with $r_{1p} = r_{1q} = 2$ and $r_{1j} = 1 j \neq 1, p, q$ is NP-hard.*

Proof. Denote by SND1 the problem defined in Theorem 5.4. We reduce SND1 to this problem. Given an instance of SND1 with n nodes add a new node 1 with arcs $(1, p)$ and $(1, q)$. (There are no other arcs touching node 1.) Call this problem (with $r_{1p} = r_{1q} = 2$ and $r_{1j} = 1$ if $j \neq 1, p, q$.) SND2. We show that there is a solution to SND1 with n arcs if and only if there is a solution to SND2 with $n + 2$ arcs.

Any solution to SND2 must use the two new arcs. Since SND2 requires two $1 - p$ paths, the other path consists of $(1, q)$ and a $q - p$ path. Similarly for q. Also SND2 requires a $1 - j$ path for every $j \neq 1, p, q$. Each such path uses either $(1, q)$ or $(1, p)$. In either case, there is a $p - j$ and a $q - j$ path, possibly using the $q - p$ or $p - q$ paths which must exist in the solution. Thus, a solution to SND2 with $n + 2$ arcs consists of a solution to SND1 with n arcs and the arcs $(1, p)$ and $(1, q)$.

The converse is obvious since any solution to SND1 augmented by the two arcs $(1, p)$ and $(1, q)$ yields a solution to SND2. ■

We note that without the connectivity requirement $r_{1j} = 1$ if $j \neq 1, p, q$, some variations are polynomially solvable, while others have been shown to be NP-complete. McCormick [17] proved that the problem $r_{s_1 t_1} = r_{s_2 t_2} = k$ ($r_{ij} = 0$ otherwise) on a directed graph is NP-hard, when k is part of the input. When $k = 1$ Li, McCormick, and Simchi-Levi [16] showed that the problem is polynomially solvable, and Natu and Fang [20,21], and Feldman and Ruhl [6] describe more efficient solutions. The latter authors also proved that the problem is polynomially solvable for any constant k. A simple modification of the reduction in [16] can be used to show that the problem is hard even when $r_{s_1 t_1} = k$ and $r_{s_2 t_2} = 1$. The reduction also applies when $s_1 = t_2$ and $s_2 = t_1$.

Theorem 5.4 showed that the SND problem with $r_{pj} = 1$ for all nodes $j \neq p$, and $r_{qp} = 1$ is NP-hard. We note that under the triangle inequality assumption, it is trivial to compute a 2-approximation for this problem. Simply compute T, an optimal directed spanning tree rooted at p (that is, satisfying $r_{pj} = 1$ for every j) and add to it the arc (q, p). Clearly $T \leq opt$ and by the triangle inequality also $w_{qp} \leq opt$.

Theorem 5.5 showed that the SND problem with $r_{1p} = r_{1q} = 2$ and $r_{1j} = 1$ $j \neq p, q$ is NP-hard. Under the triangle inequality assumption, there is a simple 2-approximation also for this problem. We first compute T, an optimal solution to the directed 2-path tree problem with nodes 1 and p. If T does not contain the arc $(1, q)$ then we return $T \cup \{(1, q)\}$. Otherwise, let $j = \arg\min\{w_{1j} + w_{jq} : j \neq 1, q\}$. If $(1, j) \in T$ we return $T \cup \{(j, q)\}$. Otherwise, we return $T \cup \{(1, j), (j, q)\}$. A feasible solution contains two $1 - q$ paths, one of which may be the arc $(1, q)$. Therefore, by the triangle inequality, $opt \geq w_{1j} + w_{jq}$. It is easy to see that the solution is feasible and since the weight of T is a lower bound on opt and also the added arc weight is at most opt, the result is a 2-approximation.

Melkonian and Tardos [18] give a factor 2-approximation for the directed SND problem when the requirement function f is crossing supermodular, i.e., for every $A, B \subset V$ such that $A \cap B \neq \emptyset$ and $A \cup B \neq V$ we have $f(A) + f(B) \leq f(A \cap B) + f(A \cup B)$. (Recall, the minimum number of arcs leaving S is defined to be $f(S)$, the requirement.) We note that f for the SND problem with $r_{1p} = r_{1q} = 2$, $r_{1j} = 1 j \neq 1, p, q$ is not crossing supermodular. Let $A = \{1, p\}$, and $B = \{1, q\}$. Then $f(A) = f(B) = 2$, $f(A \cap B) = f(\{1\}) = 2$, but $f(A \cup B) = f(\{1, p, q\}) = 1$ violating the supermodularity. Similarly, for the SND with $r_{qp} = 1$ and $r_{pj} = 1$ for all $j \neq pf$ is not crossing supermodular. Let $A = \{p, a\}$ and $B = \{q, a\}$ for some $a \neq p, q$. Then $f(A) = f(B) = f(A \cup B) = 1$ but $f(A \cap B) = f(\{a\}) = 0$.

6. Concluding remarks

We extended the range of network design problems that can be solved in polynomial time. The next simplest problem involves in the undirected case $r_{ij} = 2$ for $i, j \in S$ where $|S| = 3$, and $r_{ij} = 1$ otherwise, for which we have given an approximation algorithm. However, we do not know whether the problem is NP-complete or polynomial.

Another open problem is to extend our approximation results. For example, given two sets of nodes $S, T \subset V$ (not necessarily disjoint), we wish to have connectivity $r_{i,j} = 2$ for $i, j \in S$, $r_{i,j} = 3$ for $i, j \in T$, and $r_{i,j} = 1$ otherwise. The goal would be to give bounds better than 2, which was obtained by Jain [14]. We illustrated possible applications of the k-path tree matroid by giving approximation algorithms for two cases: $|S| \leq 5$ and $T = \emptyset$ and for the case $S = \emptyset |T| = 3$. Techniques similar to those of Section 4 may yield such algorithms.

For the directed graph case one could ask about the approximability without the assumption of triangle inequality, and whether a better factor exists with this assumption.

Acknowledgements

We thank Klaus Truemper for the proof of Theorem 2.1 which simplifies our original proof. The first author was partially supported by the NSF grant CCF-0431030.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, New Jersey, 1993.
[2] A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Designing hierarchical survivable network, Operations Research 46 (1998) 116–136.
[3] A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Intuitive solution-doubling techniques for worst-case analysis of some survivable network design problems, Operations Research Letters 29 (2001) 99–106.
[4] A. Balakrishnan, T.L. Magnanti, P. Mirchandani, Connectivity-splitting models for survivable network design, Networks 43 (2004) 10–27.
[5] C. Brezovec, G. Cornuejols, F. Glover, A matroid algorithm and its application to the efficient solution of two optimization problems on graphs, Mathematical Programming 42 (1988) 471–487.
[6] J. Feldman, M. Ruhl, The directed Steiner tree problem is tractable for a constant number of terminals, in: Proceedings 40th IEEE Annual Symposium on Foundations of Computer Science, October 17–19, 1999, New York City, NY, 1999, pp. 299–308.
[7] S. Fortune, J. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem, Theoretical Computer Science 10 (1980) 111–121.
[8] A. Frank, Kernel systems of directed graphs, Acta Scientiarum Mathematicarum 41 (1979) 63–76.
[9] G.N. Frederickson, M.A. Srinivas, Algorithms and data structure for an expanded family of matroid intersection problems, SIAM Journal on Computing 18 (1989) 112–138.
[10] F. Glover, D. Klingman, Finding minimum spanning trees with a fixed number of links at a node, in: B. Roy (Ed.), Combinatorial Programming: Methods and Applications, D. Reidel Publishing, Dordrecht, Holland, 1975, pp. 191–201.
[11] M. Grötschel, C.L. Monma, M. Stoer, Design of survivable networks, in: M.O. Ball, C.L. Monma, G. Nemhauser (Eds.), Handbook in Operations Research and Management Science, vol. 7, North-Holland, Amsterdam, 1995, pp. 617–671.
[12] M. Held, R.M. Karp, The traveling-salesman problem and minimum spanning trees, Operations Research 18 (1970) 1138–1162.
[13] H. Kerivin, A.R. Mahjoub, Design of survivable networks: A survey, Networks 46 (2005) 1–21.
[14] K. Jain, A factor 2 approximation algorithm for the generalized Steiner network problem, Combinatorica 21 (2001) 39–60.

[15] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.
[16] C.-L. Li, T.M. McCormick, D. Simchi-Levi, The point-to-point delivery and connection problems: Complexity and algorithms, Discrete Applied Mathematics 36 (1992) 267–292.
[17] S.T. McCormick, Min-cost subgraph allowing k disjoint paths from two terminals is NP-hard, Unpublished manuscript, January 2000.
[18] V. Melkonian, E. Tardos, Algorithms for a network design problem with crossing supermodular demands, Networks 43 (4) (2004) 256–265.
[19] C.St.J.A. Nash-Williams, An application of matroids to graph theory, in: P. Rosenstiel (Ed.), Theory of Graphs, Gordon and Breach, New York, 1967.
[20] M. Natu, S.-C. Fang, On the point-to-point connection problem, Information Processing Letters 53 (1995) 333–336.
[21] M. Natu, S.-C. Fang, The point-to-point connection problem — analysis and algorithms, Discrete Applied Mathematics 78 (1997) 207–226.
[22] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Springer, 2003.
[23] K. Steiglitz, P. Weigner, D.J. Kleitman, The design of minimum-cost survivable networks, IEEE Transactions on Circuit Theory 16 (1969) 455–460.
[24] A. Westerlund, M. Göthe-Lundgre, T. LArsson, A note on relatives to the Held and Karp 1-tree problem, Operations Research Letters 34 (2006) 275–282.