International Journal of Operations Research

International Journal of Operations Research Vol. 4, No. 3, 165-171 (2007)

Multicommodity Disconnecting Set Problem

Y. P. Aneja* **and X. Ke**

Odette School of Business, University of Windsor, Ontario, Canada

Received May 2006; *Revised November 2006*; *Accepted January 2007*

Abstract—Given a directed network $G = (V, A)$ with positive capacity for each $a \in A$, and a specified set of source-sink pairs of vertices, the objective is to remove a set of arcs with minimum capacity so that the resulting network stops all communication from sources to their respective sinks. We study the facial structure of the polytope associated with the solutions of this problem and identify a general class of facets. We develop two algorithms: a simple cutting plane algorithm and a branch-and-cut algorithm for this problem and present computational results.

Keywords—Multi-commodity network flows, Multi-cuts, Set covering problem, Integer programming, Polytope, Facets, Branch-and-cut, Approximation algorithms

1. INTRODUCTION

A multicommodity disconnecting set (MDS) problem, also known as a directed multicut problem, consists of a directed network $G = (V, A)$ with positive integer capacity c_a for each $a \in A$, and a set $K \subset V \times V$ of ordered pairs of nodes of G. The objective is to find a minimum capacity MDS, that is a minimum capacity arc set $C \subset A$ such that in the graph $G' = (V, A \backslash C)$, there is no (s, t)-path for any $(s, t) \in K$. Assume $|V| = n$, $|A| = m$, $A = \{a_1, a_2, ..., a_m\}$, $c_j \equiv c_{aj}$, $|K| = q$, and $K = \{(s_1, t_1), ..., (s_q, t_q)\}$.

When $q = 1$, the problem is the well known min-cut problem for single commodity networks and can be solved efficiently by finding a max-flow, and using duality to find a min-cut of the same value. This duality relationship of max-flow min-cut is generally not retained for multicommodity networks. The problem is *NP*-hard for $q \geq 3$, since a special case of this problem, the multiterminal cut problem where the objective is to disconnect a set of q nodes from each other, is *NP*-complete for $q \geq 3$ in Dahlhaus et al. (1994). For general q, the problem is *NP*-hard even on trees of height 1 (i.e. star networks) in Vazirani (2003).

The MDS problem was studied initially in the seventies (Bellmore and Ratliff (1971)) in the defense of communi-cation networks. If one considers c_j as the cost of destroying arc a_j then the attacker would want to find a minimum MDS to be able to make the communication network completely useless with minimum cost. To the best of our knowledge, the only exact algorithms for the problem in the literature were reported in Aneja and Vemuganti (1977) and Bellmore et al. (1970).

The multicut problem on undirected graphs is one of the fundamental *NP*-hard problems studied extensively for developing approximation polynomial algorithms in Shmoys (1997). The dual of the LP-relaxation of MDS-problem is the well known linear multicommodity max-flow problem in Ahuja et al. (1993). Several other applications of the problem are mentioned in Kortsarts et al. (2005), where an $O(n^{2/3})$-approximation algorithm is given for the problem. In fact most of the recent results are for developing approximation algorithms for undirected networks in Kortsarts et al. (2005). The best known approximation algorithm for the undirected networks in Garg et al. (1996) is of factor $O(\log q)$.

In this paper, we develop a general class of facets for the 0/1 polytope associated with the problem. We develop a branch-and-cut algorithm for the problem and compare its computational performance with a simple cutting plane algorithm based on approximation algorithm for solving the multicommodity max-flow problem.

2. AN INTEGER PROGRAMMING FORMULATION

Let Ω_j be the set of all simple paths (no repeated nodes) from s_k to t_k, $k = 1, ..., q$, and $\Omega = \bigcup_{j=1}^{q} \Omega_j$. Define, for each arc $a_j \in A$, a binary variable x_j which takes the value 1 if arc a_j belongs to the MDS; and 0, otherwise. Then the minimum MDS problem can be formulated in Aneja and Vemuganti (1977) as a set-covering problem:

$$\text{Minimize} \sum_{j=1}^{m} c_j x_j$$

subject to:

$$\sum_{j:a_j \in P} x_j \geq 1 \text{ for all } P \in \Omega \qquad (1)$$

$$x_j = 0/1, j = 1, ..., m.$$

If $x^* = (x_1^*, ..., x_m^*)$ is an optimal solution to Eq. (1), then $D^* = \{a_j : x_j^* = 1\}$ is called a minimum MDS.

* Corresponding author's email: aneja@uwindsor.ca

An equivalent formulation that is easy to work with, from a polyhedral study point of view, is obtained by defining a complement variable $y_j = 1 - x_j$ for all j:

Maximize $\sum_{j=1}^{m} c_j y_j$

subject to:

$$\sum_{j:a_j \in P} y_j \le |P| - 1 \text{ for all } P \in \Omega \tag{2}$$

$$y_j = 0/1 \qquad j = 1, ..., m.$$

Again, if x^* is optimal to Eq. (1), then, clearly, $y^* = 1 - x^*$ is an optimal solution to Eq. (2), where 1 is the m-vector of all ones.

We will assume, without loss of generality, that the network does not contain any arc that connects a source s_i to its sink t_i.

Let P and P' be the polytopes defined, respectively, by the convex hull of feasible solutions to Eq. (1) and Eq. (2). Then clearly there is a one-to-one correspondence between the points in P and points in P'.

Lemma 1. Polytope P is full dimensional.

Proof. Since, by assumption, no arc in the network connects any source to its sink, for any arc $a_j \in A$, the solution y such that $y_j = 1$ if $a = a_j$, and 0 otherwise, is a feasible solution to Eq. (2). Hence P' contains all m unit m-vectors and, of course, the zero m-vector. These $m + 1$ vectors are clearly affinely independent. Hence P', and, therefore, P are m dimensional.

Lemma 2. Consider a path $P \in \Omega$. Suppose no subpath of P is a path in Ω. Then the inequality $\sum_{j:a_j \in P} x_j \ge 1$ is a facet of P.

Proof. We need to show m affinely independent solutions to Eq. (1) for which this inequality holds as an equality. We will show, equivalently, that corresponding m solutions to Eq. (2) satisfy $\sum_{j:a_j \in P} y_j = |P| - 1$, and are affinely independent. Suppose path $P = (a_{i_1}, a_{i_2}, ..., a_{i_r})$, so that $|P| = r$. By assumption, $D_t = A - \{a_{i_1}, a_{i_2}, ..., a_{i_r}\} + \{a_{i_t}\}$ is an MDS for $t = 1, ..., r$. That is, removing from the network all arcs except the ones in the set $\{\{a_{i_1}, a_{i_2}, ..., a_{i_r}\} - \{a_{i_t}\}\}$ destroys all paths in Ω. This means that for $t = 1, ..., r$, the incidence vector of $A \backslash D_t$, i.e. the m-vector y^t, where:

$$y_j^t = \begin{cases} 1, \text{ if } j \in A \setminus D_t = \{a_{i_1}, a_{i_2}, ..., a_{i_r}\} - \{a_{i_t}\} \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

is a feasible solution to Eq. (2), and $\sum_{j:a_{i_j} \in P} y_j^t = r - 1$. It is quite easy to see that the r vectors $\{y^t: t = 1, ..., r\}$ are linearly, and hence affinely, independent. We now show that each arc that is not part of this path yields one additional linearly independent vector satisfying the above constraint at equality.

Consider now an arc $a_s \in A - \{a_{i_1}, a_{i_2}, ..., a_{i_r}\}$, from node u_s to node v_s. Suppose there is a proper subpath $\hat{P}$, of P, from node u_s to node v_s. Let a_{i_l} be any arc in $P \backslash \hat{P}$. Then clearly $D_s = A - \{a_{i_1}, a_{i_2}, ..., a_{i_r}\} - \{a_s\} + \{a_{i_l}\}$ is an MDS. Otherwise, if there is no such subpath $\hat{P}$, then $D_s = D_t \cup \{a_s\}$ is an MDS for any $t = 1, ..., r$. Let y^s be the incidence vector of $A \backslash D_s$. It is easy to check that y^s is feasible to (2) and $\sum_{j:a_{i_j} \in P} y_j^s = r - 1$. Now, it is quite easy to check that $S = \{y^1, ..., y^m\}$ is a linearly independent set of vectors.

Before we describe a new class of facet defining inequalities, somewhat similar to, but more general than, the Möbius Ladder inequalities defined for the "Linear Order Polytope" in Grötschel et al. (1985) and "Acyclic Subgraph Polytope" in Grötschel et al. (1985) problems, we illustrate these inequalities with a simple example generally given to show that the single commodity max-flow min-cut theorem does not hold for multi-commodity networks.

Consider a three commodity network in Figure 1. This network has three chains with one chain for each commodity. Setting all arc capacities to one, an optimal disconnecting set is obtained by removing any two arcs from the set $\{a_1, a_2, a_3\}$, with a total capacity of 2. Now solving the relaxed linear program (LP) associated with (1) gives the following linear program:

Minimize $\sum_{j=1}^{9} x_j$

subject to:

$$\begin{aligned} &x_4 + x_1 + x_2 + x_9 \ge 1 \\ &x_6 + x_2 + x_3 + x_5 \ge 1 \\ &x_8 + x_3 + x_1 + x_7 \ge 1 \\ &x_j \ge 0 \ \forall j = 1, ..., 9. \end{aligned} \tag{4}$$

An optimal solution to the above linear program is given by x^* where $x_1^* = x_2^* = x_3^* = 1/2$, and $x_j^* = 0, j = 4, ..., 9$, otherwise, resulting in an optimal objective value of 3/2. We derive a Chvatal-Gomory (C-G) inequality in Wolsey (1998) by adding up the three constraints and dividing by two:

$$x_1 + x_2 + x_3 + (1/2)\sum_{j=4}^{9} x_j \ \ge 3/2$$

which implies: $\sum_{j=1}^{j=9} x_j \ge 3/2$, and hence $\sum_{j=1}^{9} x_j \ge 2$, as x is integer. Adding this constraint to the LP (4) and solving the problem results in an integer optimal solution with an objective value of 2.

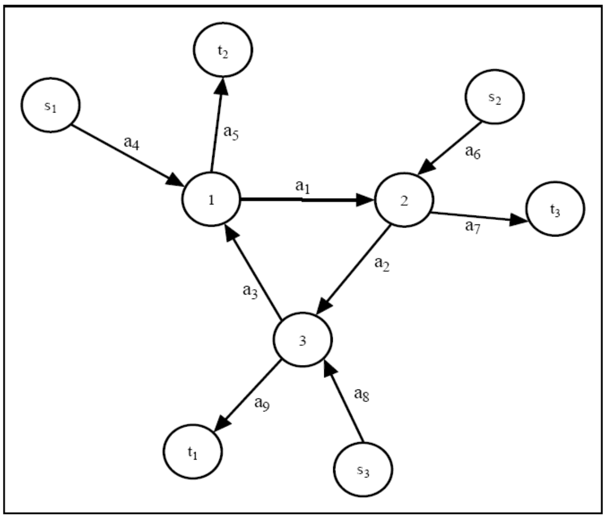

Figure 1. A three commodity network.

Here is another example of four commodity sub-network in Figure 2:

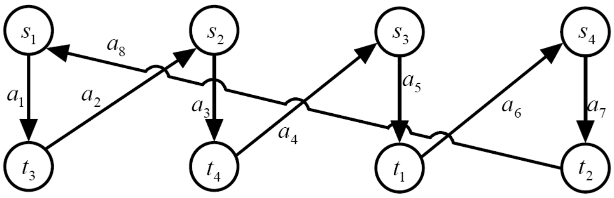

Figure 2. A 4-commodity network.

Assuming all arcs with unit capacity, the relaxed LP is:

Minimize $\sum_{j=1}^{8} x_j$

subject to:

$$\begin{aligned} &x_1 + x_2 + x_3 + x_4 + x_5 \geq 1 \\ &x_3 + x_4 + x_5 + x_6 + x_7 \geq 1 \\ &x_1 + x_5 + x_6 + x_7 + x_8 \geq 1 \\ &x_1 + x_2 + x_3 + x_7 + x_8 \geq 1 \\ &x_j \geq 0 \ \forall j = 1, ..., 8, \end{aligned} \tag{5}$$

whose optimal solutions is x^*, where $x_j^* = 1/5$ for all j. We get a C-G valid inequality by adding these four inequalities, dividing by 3: $\sum_{j=1}^{8} x_j \geq 2$. Solving LP (5), with this additional inequality, yields an integer optimal solution with $x_1^* = x_3^* = 1$, and $x_j^* = 0$, otherwise.

It will follow from our lemma below that such C-G valid inequalities are indeed facets of the polytope P.

Definition 1. Let $S = \{P_1, P_2, ..., P_k\}$ be a set of k different paths in Ω, such that no subpath of any of these paths is a path in Ω. Let r be an integer such that $2 \leq r \leq k - 1$ and r does not divide k. For convenience, assume that for $s > k$, P_s would refer to the path $P_{s \bmod k}$. Let M be the subgraph obtained by the union of the k paths in S. Then M is called a Möbius Ladder if:

1. Every r "consecutive paths" of S have a common subpath. That is, paths P_i, P_{i+1}, ..., P_{i+r-1} have a common subpath $\hat{P}_i$, $i = 1, ..., k$.
2. Subpaths $\hat{P}_i$, $i = 1, ..., k$ are mutually disjoint.
3. Any non-consecutive r paths of S have no subpath in common.

Note that Figure 1 and Figure 2 are Möbius Ladders with $k = 3$, $r = 2$, and $k = 4$, $r = 3$, respectively.

Lemma 3. If M is a Möbius Ladder then $\sum_{a_j \in M} x_j \geq \left\lceil \frac{k}{r} \right\rceil$ is a facet of polytope P.

Proof. We first show that this is a valid inequality for P. Let B be the set of all arcs in subpaths $\hat{P}_1$, $\hat{P}_2$, ..., $\hat{P}_k$. That is $B = \{ a_j : a_j \in \bigcup_{i=1}^{i=k} \hat{P}_i \}$. Consider addition of k valid path inequalities corresponding to the k paths in S:

$$\sum_{i=1}^{k} \sum_{a_j \in P_i} x_j \geq k \tag{6}$$

This inequality can be rewritten as:

$$\sum_{j: a_j \in B} \alpha_j x_j + \sum_{j: a_j \in M \setminus B} \alpha_j x_j \geq k \tag{7}$$

From properties 1. and 2. of the Möbius Ladder, it follows that $\alpha_j = r$ for j such that $a_j \in B$, and from property 3. of the Möbius Ladder, we can conclude that $1 \leq \alpha_j < r$, for j such that $a_j \in M \backslash B$. Hence, dividing Eq. (7) by r, we get:

$$\sum_{j: a_j \in B} x_j + \sum_{j: a_j \in M \setminus B} \frac{\alpha_j}{r} x_j \geq \frac{k}{r} \tag{8}$$

which implies:

$$\sum_{j: a_j \in B} x_j + \sum_{j: a_j \in M \setminus B} x_j \geq \frac{k}{r} \tag{9}$$

Since the left hand side is an integer, we get the C-G inequality:

$$\sum_{a_j \in M} x_j \geq \left\lceil \frac{k}{r} \right\rceil \tag{10}$$

Hence, Möbius Ladder inequalities are valid for P.

We now show that each arc in M provides an MDS for G whose incidence vector x satisfies Eq. (10) at equality, and that the $|M|$ incidence x-vectors so obtained are linearly independent. For a path P_j from the set S, consider the following set of $\left\lfloor \frac{k}{r} \right\rfloor$ subpaths: $\{\hat{P}_{j+1}$,

$\widehat{P}_{j+r+1}, ..., \widehat{P}_{j+(\lfloor \frac{k}{r} \rfloor -1)r+1}\}$. Then removing one arc from each one of these disjoint subpaths, and any arc from P_j, and all arcs of $A\backslash M$, provides an MDS of cardinality $\lceil \frac{k}{r} \rceil$ + $|A\backslash M|$, and hence its incidence x-vector satisfies Eq. (10) as an equality. It is easy to see that the $|P_j|$ incidence x-vectors obtained in this manner are linearly independent. Proceeding in this manner with each of the k paths in S, we can obtain $|M|$ such incidence x-vectors which are linearly independent.

Equivalently, we have $|M|$ complement linearly independent y-vectors which satisfy $\sum_{a_j \in M} y_j = |M| - \lceil \frac{k}{r} \rceil$. Each such y-vector is an incidence vector of a disconnected graph—the graph after an MDS is removed from the graph, and hence contains all arcs of the Möbius Ladder except the $\lceil \frac{k}{r} \rceil$ arcs removed as described above.

Finally we prove the lemma by providing $|A\backslash M|$ additional such linearly independent y-vectors, one for each arc in $A\backslash M$, which satisfy $\sum_{a_j \in M} y_j = |M| - \lceil \frac{k}{r} \rceil$. Arguments are similar to the ones given in Lemma 1. Consider an arc $a_s \in A\backslash M$. There are two cases to consider. In the first case, suppose a_s is an arc from node u_s to node v_s, and there is a subpath $\widehat{P}$, of $P_j \in S$, from node u_s to node v_s in M. Then, consider the following disconnected set of arcs: remove from $M \cup \{a_s\}$ an arc from $P\backslash \widehat{P}$, and one arc from each of the $\lfloor \frac{k}{r} \rfloor$ subpaths: $\{\widehat{P}_{j+1}$, $\widehat{P}_{j+r+1}$, ..., $\widehat{P}_{j+(\lfloor \frac{k}{r} \rfloor -1)r+1}\}$. Incidence y-vector of this disconnected set clearly satisfies $\sum_{a_j \in M} y_j = |M| - \lceil \frac{k}{r} \rceil$. In the second case, adding the arc a_s does not add any subpath to M, and any disconnected network obtained earlier, with this additional arc, still remains disconnected. Further, its incidence y-vector satisfies the Möbius Ladder equality and is clearly linearly independent of the existing $|M|$ y-vectors and any other y-vector obtained in this manner for other arcs in $A\backslash M$. This proves our lemma.

From an algorithmic point of view it is important to know if we can efficiently solve the separation problem for path as well as Möbius Ladder inequalities. We do not know of any efficient method for identifying violated Möbius Ladder inequalities—i.e. an efficient algorithm for testing if a given fractional solution to the relaxed LP associated with Eq. (1) violates a Möbius ladder inequality. But Dijkstra's shortest path algorithm in Dijkstra (1959) solves efficiently the separation problem for path inequalities.

Before we describe a branch-and-cut algorithm for our problem, we shall review certain results for a cutting plane algorithm studied in Aneja and Vemuganti (1977). The cuts developed in Aneja and Vemuganti (1977) will be used, along with path-cuts, in our branch-and-cut algorithm. Also, we will describe a simple modification of this cutting plane algorithm, that does not require us to solve the relaxed LP associated with Eq. (1) exactly, and compare its computational performance with the branch-and-cut algorithm.

Consider the relaxed LP associated with the set covering problem Eq. (1) in a matrix form:

Minimize $c^{\mathrm{T}}x$
subject to:

$$Ax \geq 1 \qquad (11)$$
$$x \geq 0.$$

Matrix A is the path-arc incidence matrix of G and is known only implicitly. Any 0/1 solution of Eq. (11) is the incidence vector of an MDS. If we define a "proper MDS" as an MDS such that none of its proper subsets is an MDS, then the incidence vector of any proper MDS is an extreme point of Eq. (11). Further, the rows of $\overline{A}$, the submatrix of A formed by columns of a proper MDS with cardinality p, can be permuted to obtain $\overline{A}^P$ such that the top p rows of $\overline{A}^P$ form an identity matrix. Specifically, suppose $D_0 = \{a_{i_1}, ..., a_{i_p}\}$ is a proper MDS, and $\overline{A} = (A_{i_1}, ..., A_{i_p})$, then there exists a permutation matrix P such that $P\overline{A} = \overline{A}^P = \begin{bmatrix} I \\ R \end{bmatrix}$. A basis B_0 for the corresponding extreme point can be easily obtained by adding surplus variables corresponding to rows of R. Clearly $PB_0 = B_0^P = \begin{bmatrix} I & 0 \\ R & -I \end{bmatrix}$, $B_0^P B_0^P = I$, and $B_0^{-1} = B_0^P P$. If we define, $c_{B_0}^{\mathrm{T}} = (c_{i_1}, ..., c_{i_p} | 0, ..., 0) = (c_{B_1}^{\mathrm{T}}, 0)$, $A^P = PA$, and $A_j^P = PA_j$, then it is easy to see that:

$$z_j = c_{B_0}^{\mathrm{T}} B_0^{-1} A_j = c_{B_0}^{\mathrm{T}} B_0^P P A_j = (c_{B_0}^{\mathrm{T}} B_0^P) A_j^P = c_{B_1}^{\mathrm{T}} \overline{A}_j^P$$

Hence, the reduced cost corresponding to x_j is given by $\overline{c}_j = c_j - c_{B_1}^{\mathrm{T}} \overline{A}_j^P$.

That is, reduced costs for all non-basic variables can be obtained by knowing only the top p rows of A^P. Further, to obtain these top p rows, one for each arc in D_0, all we need is the corresponding proper MDS D_0. To generate a row corresponding to arc a_{i_1} we proceed as follows. Obtain $\widehat{G}$ by removing from G all arcs of D_0 except a_{i_1}. Any path $P \in \Omega$ that is still a path in $\widehat{G}$ must use arc a_{i_1} since D_0 is a proper MDS, and hence can be used as a row corresponding to arc a_{i_1}. Remaining $(p - 1)$ rows are generated similarly.

So given D_0, define Q to be the set of non-basic variables with negative reduced costs. That is, $Q = \{j$: $\overline{c}_j$

$< 0\}$. Then $Q = \emptyset$ implies that D_0 is an optimal MDS. Equivalently, if D_0 is not optimal then $Q \neq \emptyset$, and an optimal MDS must satisfy the cut $\sum_{j \in Q} x_j \geq 1$. Based on this result, the following cutting plane algorithm was proposed in Aneja and Vemuganti (1977) for Eq. (1):

Algorithm 1.

Step 1. Denote the original problem Eq. (1) as SC_1, $\bar{z} = \sum_{j=1}^{m} c_j$, and $t = 1$.

Step 2. Let $\hat{x}^t$ be an optimal solution to the relaxed LP associated with SC_t. Terminate if $\sum_{j=1}^{m} c_j \hat{x}_j^t \geq \bar{z}$. Otherwise, $\lceil \hat{x}^t \rceil$ is a feasible solution to SC_t. Extract from $\lceil \hat{x}^t \rceil$ a proper solution x^t to SC_t. If $c^T x^t < \bar{z}$, set $\bar{z} = c^T x^t$, and record x^t as the best solution so far.

Step 3. Determine Q, the set of variables with negative reduced cost. Terminate if $Q \neq \emptyset$. Other create SC_{t+1} by adding the cut $\sum_{j \in Q} x_j \geq 1$ to SC_t, set $t \leftarrow t + 1$ and go back to *Step* 2.

Computational experiments with the algorithm reported in Aneja and Vemuganti (1977) showed that in *Step* 2, either the LP provided an integer optimal solution, or strong lower bound provided by the LP solution was crucial to termination of the algorithm. To obtain an optimal solution to the relaxed LP associated with SC_1, note that the dual of the relaxed LP is the multicommodity max-flow problem:

$$\text{Maximize} \sum_{P \in \Omega} f_P$$

subject to:

$$\sum_{P: P \in \Omega, a_j \in P} f_P \leq c_j \tag{12}$$

$$f_P \geq 0, \forall P \in \Omega.$$

Columns of Eq. (12) are known only implicitly. This LP was solved by the revised simplex method using a column generation scheme. Clearly, the dual optimal solution provided $\hat{x}^1$. With slight modification $\hat{x}^t$ is similarly obtained.

We are now ready to describe our modified simple cutting plane algorithm. Recently a very simple and interesting algorithm has been proposed in Garg and Könemann (1998) and improved in Fleischer (2000) to obtain an ε-approximate solution to Eq. (12). This fully polynomial approximation scheme (FPAS) terminates with feasible solutions to both the primal and the dual LPs, and proceeds as follows.

The algorithm starts with zero flow and $x_j = \delta$ for each arc a_j in A, where $\delta = (1+\varepsilon)(n(1+\varepsilon))^{-1/\varepsilon}$. The algorithm proceeds in iterations. Treating x_j as the length of arc a_j, it finds a shortest path, say P_0, in Ω. Let $c_0 = \min\{c_j: a_j \in P_0\}$. Flow on path P_0 is augmented by c_0, and length of each arc in P_0 is modified as follows: $x_j \leftarrow x_j(1 + \varepsilon(c_0/c_j))$ $\forall a_j \in P$. This process of finding a shortest path and augmenting flows is repeated until the length of any shortest path in Ω is at least 1. At this point appropriate scaling of flows and x-value of each arc provides the desired pair of primal and dual solutions. The algorithm terminates in at most $(m/\varepsilon)(\log_{1+\varepsilon} n)$ iterations.

We modified the cutting plane scheme given in Algorithm 1 by replacing the LP optimal solutions with the ones obtained by the FPAS scheme described above, with some simple modifications. For example, we use the approximate solution to the relaxed LP for SC_t as a starting solution for the solving relaxed LP for SC_{t+1}.

For the branch-and-cut algorithm we proceed, informally, as follows. We start with solving an LP with $|K| = q$ constraints: one path-constraint for each $(s, t) \in K$. Formally, let $P_1, \ldots, P_q$ be any q simple paths from $(s_1, t_1), \ldots, (s_q, t_q)$, respectively. Solve the LP:

$$\text{Minimize} \sum_{j=1}^{m} c_j x_j$$

subject to:

$$\sum_{a_j \in P_i} x_j \geq 1, i = 1, \ldots, q \tag{13}$$

$$x_j \geq 0, j = 1, \ldots, m.$$

Let x^0 be an optimal solution to Eq. (13). Treating x_j^0 as the length of arc a_j, find a shortest path, say P_0 with length $l(P_0)$, in Ω. If $l(P_0) < 1$, we add the cut $\sum_{a_j \in P_0} x_j \geq 1$ to Eq. (13) and resolve the LP. We repeat this process as long as the shortest path length is less than one.

Suppose this process stops with an optimal LP solution $\hat{x}^0$. If $\hat{x}^0$ is integer, we terminate as we have found an optimal solution to Eq. (1). Otherwise, we consider the 0/1 vector $\lceil \hat{x}^0 \rceil$ which is a feasible solution to Eq. (1). The corresponding MDS is $\widehat{D} = \{a_j: \hat{x}_j^0 > 0\}$. To obtain a "good" proper MDS D^0 from $\widehat{D}$, we sort the arcs of $\widehat{D}$ in non-decreasing order of their x-values, and try removing each arc from $\widehat{D}$, in that order, to see if the resulting set is still an MDS. Using D^0 we generate a set Q as described earlier. If $Q = \emptyset$, we terminate with an optimal MDS D^0.

Suppose $Q \neq \emptyset$. If $\sum_{j \in Q} \hat{x}_j^0 < 1$, the cut $\sum_{j \in Q} x_j \geq 1$ is added to the existing LP and resulting LP is resolved, repeatedly, until the shortest path length is at least one. If $\sum_{j \in Q} \hat{x}_j^0 \geq 1$, we branch on a variable x_j with fractional $\hat{x}_j^0$-value, and proceed with the branch-and-bound algorithm in a standard manner.

Note that at any intermediate node of the branch-and-bound tree, variables can be partitioned into three sets: S^1 (variables set to 1), S^0 (variables set to 0), and F (free variables). Clearly solving Eq. (11) with these restrictions is equivalent to solving Eq. (11) over a network where the arcs corresponding to variables in S^1 are removed from the network, and an infinite cost is assigned to arcs for variables in S^0. This allows us to use path-cuts as well as Q-cuts at intermediate nodes of the tree as well.

3. COMPUTATIONAL EXPERIMENTS

To compare the two approaches, 140 networks were randomly generated as follows. We tried two different sets of nodes: 50 and 100; five different sets of arcs: 100, 200, 300, 400 and 500; two sets of commodities with $|K| = 5$ and 10, and either all arc capacities to be 1, or random integer from 1 to 8. The capacities range was thus from 1 to $c_{\max}$ (= 1 or 8). For each combination tried, 5 random problems were generated. Each network was generated as follows. First, a directed cycle with n nodes was created to create a strongly connected network of m arcs, and remaining $m - n$ arcs were then added at random.

Table 1 below provides summary information for 28 different sets. Both algorithms were run using Xpress-MP on a PC with Pentium IV(R) CPU 3GHz with 2GB of memory. Numbers reported in the table are for 28 sets of 5 problems each for certain node size and arc size, giving rise to a total of 140 problems in our computational experiments. For the branch-and-cut algorithm, the two columns, respectively, report the average number of cuts, and the average time in seconds for the set of 5 randomly generated problems. For the modified cutting plane algorithm, the time limit for each problem was set at 1800 seconds. Column labeled "# solved" reports the number of problems, out of a set of 5 problems, for which the algorithm stopped with an optimal solution. The "Avg. time" column represents the average time for solved problems in each set.

For the branch-and-cut algorithm, 131 of 140 problems terminated at the first iteration. Out of these 131 problems, 120 terminated with an integer LP solution, and the remaining 11 terminated either because the rounded up LP objective value was equal to the value of the proper MDS found, or the set Q generated for the MDS was a null set.

As was reported in our computational results earlier in Aneja and Vemuganti (1977), Algorithm 1, with exact LP solutions, performed well since most of the time the LP optimal solution was integer. When the LP solution did not terminate integer, and the algorithm did not terminate at the first iteration, its performance was not very satisfactory. The branch-and-cut algorithm proposed here overcomes this difficulty and performs extremely well under these circumstances. The depth of the branch-and-bound tree was rather small in most cases. In our experiments, only one tree reached a depth of 8, and no other exceeded a depth of 3. For exploring the branch-and-bound tree we tried both depth-first search (putting unfathomed nodes on a stack) and breadth-first search (putting unfathomed nodes on a queue). There was no discernible difference in the performance of the algorithm between these two strategies.

As is clear from our computational results, Algorithm 1 performed poorly. This primarily is due to poor performance of the approximation algorithm for solving linear programs. Only 115 of the 140 problems tried terminated with an optimal solution in the specified time

Table 1. Computation results

		Branch-and-Cut		Mod. Cutting Plane	
Set	$n, m, q, c_{\max}$	Avg. # cuts	Avg. time (sec)	# solved	Avg. time (sec)
1	50, 100, 5, 1	12.4	< 0.5	5	5.8
2	50, 100, 5, 8	11	< 0.5	5	6
3	50, 100, 10, 1	26.4	0.8	1	12
4	50, 100, 10, 8	22	0.5	4	23.3
5	50, 200, 5, 1	38.8	1	3	3.3
6	50, 200, 5, 8	48	1.3	5	38.8
7	50, 200, 10, 1	91.6	2	3	43.3
8	50, 200, 10, 8	69.2	1.5	5	80.8
9	50, 300, 5, 1	60.8	1	5	15.8
10	50, 300, 5, 8	69.8	1	5	74.4
11	50, 300, 10, 1	127.2	2.8	5	75.8
12	50, 300, 10, 8	159.6	4.3	3	427.3
13	100, 200, 5, 1	16.4	1.5	5	35.4
14	100, 200, 5, 8	24.6	1.8	5	71.8
15	100, 200, 10, 1	52.8	2.5	4	43.5
16	100, 200, 10, 8	51.4	4	1	130
17	100, 300, 5, 1	24.6	2	5	20.2
18	100, 300, 5, 8	22	2.3	4	38.8
19	100, 300, 10, 1	63.6	4	4	117
20	100, 300, 10, 8	82	2.5	4	491.3
21	100, 400, 5, 1	43.8	1.8	4	19.8
22	100, 400, 5, 8	71.4	2.3	4	112.8
23	100, 400, 10, 1	90.4	3.3	5	112.8
24	100, 400, 10, 8	114.8	3.5	3	497.3
25	100, 500, 5, 1	45.2	3.3	5	28.4
26	100, 500, 5, 8	60	4.5	5	152.8
27	100, 500, 10, 1	110.4	10.8	5	258.4
28	100, 500, 10, 8	130.8	12	3	1033

limit of 30 minutes. Of the remaining 25 problems, 15 problems had generated the optimal solution as an upper bound.

4. CONCLUSIONS

In this paper we have presented two algorithms for the multicommodity disconnecting set problem: a branch-and-cut algorithm, and a simple cutting plane algorithm that relies on solving linear programs approximately by applying Dijkstra's shortest path algorithm repeatedly. From a computational point of view, the branch-and-cut algorithm performs extremely well and is far superior to the cutting plane algorithm.

The advantage of our simple cutting plane algorithm of course lies in its simplicity and its ability to work with much larger problems due to very minimal memory and space requirements. We are currently exploring ways in which the approximation algorithm for solving LPs could be expedited thereby improving performance of the cutting plane algorithm.

REFERENCES

1. Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (Eds.) (1993). *Network Flows*, Prentice Hall, New Jersey.
2. Aneja, Y.P. and Vemuganti, R. (1977). A row generation scheme for finding a multicommodity disconnecting set. *Management Science*, 23: 652-659.
3. Bellmore, M., Greenberg, H., and Jarvis, J. (1970). Multicommodity disconnecting sets. *Management Science*, 16: 427-433.
4. Bellmore, M. and Ratliff, H. (1971). Optimal defence of multicommodity networks. *Management Science*, 18: 174-185.
5. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., and Yannakakis, M. (1994). The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23: 864-894.
6. Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numeriche Mathematics*, 1: 269-271.
7. Fleischer, L. (2000). Approximating fractional multicommodity flows independent of the number of commodities. *SIAM Journal of Discrete Mathematics*, 13: 505-520.
8. Garg, N. and Könemann, J. (1998). Faster and simpler algorithm for multicommodity flow and other fractional packing problems. *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, Palo Alto, California, USA, pp. 300-309.
9. Garg, N., Vazirani, V., and Yannakakis, M. (1996). Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25: 235-251.
10. Grötschel, M., Junger, M., and Reinelt, G. (1985). On the acyclic subgraph polytope. *Mathematical Programming*, 33: 28-42.
11. Grötschel, M., Junger, M., and Reinelt, G. (1985). Facets of the linear ordering polytope. *Mathematical Programming*, 33: 43-60.
12. Kortsarts, Y., Kortsarts, G., and Nutov, Z. (2005). Greedy approximation algorithms for directed multicuts. *Networks*, 45: 214-217.
13. Shmoys, D.B. (1997). Cut problems and their application to divide-and-conquer. In: D.T. Hochmaum (Ed.), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, Boston.
14. Vazirani, V. (Ed.) (2003). *Approximation Algorithms*, Springer-Verlag, New York.
15. Wolsey, L.A. (Ed.) (1998). *Integer Programming*, John Wiley and Sons, New York.