RC22984 (W0311-069) November 10, 2003
Mathematics

IBM Research Report

Separation of Partition Inequalities with Terminals

Francisco Barahona
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Hervé Kerivin
Institute for Mathematics and Its Applications
University of Minnesota
357 Lind Hall
207 Church Street S.E.
Minneapolis, MN 55455

Francisco Barahona · Hervé Kerivin

SEPARATION OF PARTITION INEQUALITIES WITH TERMINALS

October 2003

Abstract. Given a graph with n nodes each of them having labels equal either to 1 or 2 (a node with label 2 is called a terminal), we consider the (1,2)-survivable network design problem and more precisely, the separation problem for the partition inequalities. We show that this separation problem reduces to a sequence of submodular flow problems. Based on an algorithm developed by Fujishige and Zhang the problem is reduced to a sequence of $O(n^4)$ minimum cut problems.

1. Introduction

In telecommunication networks some nodes may be more important than others because of their specific functions. This fact leads to specify certain survivability conditions. Thus, it is usual to consider two kinds of nodes, the *specific nodes*, also called *terminals*, for which a "high" degree of survivability has to be guaranteed and the *ordinary nodes* which simply have to be connected to the network. The network topology problem then consists of selecting links so that the sum of their cost is minimized and the failure of any single link may not disconnect any two terminal nodes.

More precisely, based on a model first introduced by Grötschel and Monma [13] (see also Stoer [29]), this problem can be stated as follows. Consider an undirected graph $G = (V, E)$ where V represents the node set, and E represents the set of edges or potential links. The set V is partitioned into two subsets T and O corresponding respectively to the terminal and ordinary nodes. By associating to each node $u \in V$, a label $r(u)$, called its *connectivity type*, which is equal to 1 if u is an ordinary node, and to 2 if u is a terminal, we get $O = \{u \in V : r(u) = 1\}$, $T = \{u \in V : r(u) = 2\}$ and $V = O \cup T$. The survivability conditions require the existence of at least $\min\{r(s), r(t)\}$ edge-disjoint paths in the subgraph of G for any pair of nodes $s, t \in V$. Such a subgraph is called *survivable*. Suppose that each edge $e \in E$ has a certain cost $c(e) \in \mathbb{R}_+$, then our network topology problem, called *survivable network design problem* and denoted SNDP, consists of finding a survivable subgraph of G with minimum total cost.

Francisco Barahona: IBM T.J. Watson Research Center, Yorktown Heights, NY 10589, USA. e-mail: barahon@us.ibm.com

Hervé Kerivin: Institute for Mathematics and its Applications (IMA), University of Minnesota, 357 Lind Hall, 207 Church Street S.E., Minneapolis, MN 55455, USA. e-mail: kerivin@ima.umn.edu

The optimization problem SNDP is NP-hard, and it has been extensively studied in the past. Some heuristics have been devised (see [24] for instance) and the SNDP has been proved to be polynomially solvable in some particular cases (see [13,23] for instance). Particularly, we point out that if $T = \emptyset$ (i.e., $r(u) = 1$ for all $u \in V$) then the SNDP is nothing but the minimum cost spanning tree problem which is well-known to be polynomially solvable [21]. For a complete survey over the existing approaches to survivability problems related to the SNDP, see Grötschel, Monma and Stoer [16] and Stoer [29]. Grötschel, Monma and Stoer [14] studied the general model related to the SNDP (i.e., $r(u) \in \mathbb{Z}_+$ for all $u \in V$) from a polyhedral point-of-view. They considered valid inequalities for the polytope associated with this problem, and they derived some necessary and/or sufficient conditions under which these inequalities are facet-defining.

Among all the inequalities considered in [14], the so-called partition inequalities have appeared to be useful for solving the general model related to the SNDP. Grötschel, Monma and Stoer [14] gave sufficient conditions and necessary conditions for the partition inequalities to be facet-defining. In [15], they showed that the separation problem for the partition inequalities is NP-hard for general connectivity types $r \in \mathbb{Z}_+^V$. Because of their computational intractability, Grötschel, Monma and Stoer [15] devised some separation heuristics for the partition inequalities which were successful in speeding up their branch-and-cut algorithm.

For the SNDP we are interested in in this paper, the partition inequalities have recently been studied more deeply. Didi Biha, Kerivin and Mahjoub [6] showed that the partition inequalities together with the trivial lower-bound and upper-bound inequalities completely describe the polytope associated to the SNDP when the graph G is series-parallel. Furthermore, Kerivin and Mahjoub [19] showed that the partition inequalities can be separated in polynomial time. However their algorithm leads to a time complexity which does not permit to implement it. Therefore, they have developed a heuristic for separation and some computational results pointing out the usefulness of the partition inequalities in a branch-and-cut algorithm for the SNDP are reported in [20].

In this paper, we study the separation problem again and improve the time complexity led by the algorithm devised by Kerivin and Mahjoub [19]. Here, we show that the separation problem reduces to a sequence of n submodular flow problems, where the complexity of solving each of them is dominated by the complexity of solving $O(n^3)$ minimum cut problems.

This paper is organized as follows. In Section 2 we review several type of partition inequalities. Section 3 is devoted to the reduction of the separation problem for partition inequalities to a sequence of submodular flow problems. Then, we show in Section 4 how Fujishige-Zhang algorithm for the submodular intersection problem applies to our problem. In Section 5 we describe how to change terminals. In Section 6 we study a related question for the case with three terminals. Finally, some concluding remarks are given in Section 7.

The rest of this section is devoted to more definitions and notations. The graphs we consider are finite, loopless and connected. We deal with an undirected graph $G = (V, E)$, if $e \in E$ is the unique edge between two nodes u and v, then we write uv to denote the edge e. If $F \subset E$ then $G' = (V, F)$ is called a *spanning subgraph*. If $W \subset V$, and $E(W)$ is the set of edges with both endnodes in W, then $G'' = (W, E(W))$ is called the *subgraph induced by* W. We denote by n the number of nodes of G, $n = |V|$. For $W \subseteq V$, the set of edges having exactly one endnode in W is called a *cut* and is denoted by $\delta(W)$. Moreover, if $s \in W$ and $t \notin W$, then $\delta(W)$ is called an *st-cut*. If $W = \{u\}$, then we write $\delta(u)$ for $\delta(\{u\})$. Given a partition $\{V_1, \dots, V_p\}$ of the node set V, we denote by $\delta_G(V_1, \dots, V_p)$ the set of edges with endnodes in two different sets $\{V_i\}$. We use $\delta(V_1, \dots, V_p)$ whenever the graph G can be deduced from the context. If $D = (V, A)$ is a directed graph and $a \in A$ is the unique arc from the node u to the node v, then we write (u, v) to denote the arc a. The tail u of the arc a is denoted by $\partial^+ a$, and its head v by $\partial^- a$.

Given a ground set S, a set-function $f : 2^S \longrightarrow \mathbb{R} \cup \{\infty\}$ is called *fully submodular* if

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \tag{1}$$

for all $A, B \subseteq S$. A pair of subsets A and B of S is said to be *intersecting* if none of $A \setminus B$, $B \setminus A$, $A \cap B$ is empty. Then a set-function f is called *submodular on intersecting pairs* if inequality (1) is required only for intersecting pairs. For a vector $x \in \mathbb{R}^S$ and a subset $A \subseteq S$, we denote $\sum_{a \in A} x(a)$ by $x(A)$. For any $u \in S$, χ_u is an element in $\mathbb{R}^S$ such that $\chi_u(u) = 1$ and $\chi_u(v) = 0$ for $v \in S \setminus \{u\}$. For $F \subseteq S$ the *incidence vector* of F, $x^F \in \mathbb{R}^S$, is defined by $x^F(e) = 1$ if $e \in F$, $x^F(e) = 0$ if $e \in S \setminus F$.

A system $Ax \leq b$ in n dimensions is called *totally dual integral* (or just *TDI*) if A and b are rational and for each $c \in \mathbb{Z}^n$, the dual of maximizing $c^T x$ over $Ax \leq b$ has an integer optimum solution y, if it is finite.

2. Partition inequalities

In this section we define several types of partition inequalities and comment on their separation algorithms.

2.1. Preliminaries

Let $G = (V, E)$ be a graph and a vector $x : E \to \mathbb{R}$. A first type of partition inequalities is

$$x\big(\delta(S_1, \dots, S_p)\big) \geq (p-1), \text{ for all partitions } \{S_1, \dots, S_p\} \text{ of } V. \tag{2}$$

It follows from [30] and [25] that these inequalities together with $x \geq 0$, define a polyhedron whose extreme points are the incidence vectors of spanning trees of G.

For a class of inequalities, the separation problem is: *given a vector $\bar{x}$ find a violated inequality in the class or prove that none exists.* An algorithm for the separation problem is a key ingredient for being able to use a class of inequalities within a branch-and-cut algorithm.

The separation problem for inequalities (2) has been studied by Cunningham [3] who reduced it to a sequence of $|E|$ minimum cut problems. Later Barahona [2] reduced it to n minimum cut problems. In both cases they solve

$$\text{minimize } \bar{x}(\delta(V_1,\ldots,V_p)) - p, \tag{3}$$

where the minimization is among all partitions of V.

A more general type of partition inequalities is

$$x\big(\delta(S_1,\ldots,S_p)\big) \geq ap + b, \text{ for all partitions } \{S_1,\ldots,S_p\} \text{ of } V, \tag{4}$$

for fixed constants a and b. The separation problem in this case was studied by Baiou, Barahona & Mahjoub [1]. This also reduces to problem (3) but depending on the values of a and b, in some cases one has to exclude the trivial partition ($p = 1$) and impose $p \geq 2$. This reduces to $O(n^3)$ minimum cut problems.

2.2. The present study

Let $G = (V,E)$ be a graph and $r \in \{1,2\}^V$ be a connectivity type vector. Let SNDP(G,r) be the convex hull of incidence vectors of survivable subgraphs. The set of ordinary nodes is denoted by $O = \{u \in V : r(u) = 1\}$ and the set of terminals by $T = \{u \in V : r(u) = 2\}$. For a nonempty node subset $W \subset V$, let $r(W) = \max\{r(u) : u \in W\}$ and $con(W) = \min\{r(W), r(V \setminus W)\}$. If (V,F) is a survivable subgraph of G, then its incidence vector x^F satisfies

$$0 \leq x(e) \leq 1 \qquad \text{for all } e \in E, \tag{5}$$

$$x(\delta(W)) \geq con(W) \qquad \text{for all } \emptyset \neq W \subset V. \tag{6}$$

The inequalities (5) and (6) are called respectively *trivial inequalities* and *cut inequalities*. The separation problem for the cut inequalities (6) is polynomially solvable using a minimum *st*-cut algorithm (e.g., preflow-push algorithm of Goldberg and Tarjan [10] running in $O(n^3)$ time).

In [14], Grötschel, Monma and Stoer introduced a class of valid inequalities for the polytope SNDP(G,r) called partition inequalities, and which can be stated as follows. Let $\{V_1,\ldots,V_p\}$, $p \geq 2$, be a partition of V. Let $I_2 = \{i : con(V_i) = 2,\ i = 1,...,p\}$ be the set of indices whose corresponding sets of the partition contain at least one terminal. The *partition inequalities* induced by $\{V_1,\ldots,V_p\}$ is

$$x(\delta(V_1,\ldots,V_p)) \geq \begin{cases} p-1 & \text{if } I_2 = \emptyset, \\ p & \text{otherwise.} \end{cases} \tag{7}$$

The inequalities (7) are a generalization of the cut inequalities (6). (This is the case where $p = 2$.) Grötschel, Monma and Stoer [14] gave sufficient conditions and necessary conditions for the inequalities (7) to define facets for SNDP(G,r).

In [19], Kerivin and Mahjoub showed that the separation problem for the partition inequalities (7) reduces to minimizing a particular submodular function, and then is polynomially solvable. Their approach can be described as follows. First of all, they consider the case where all the terminals belong to the same set of the partition, that is $I_2 = \emptyset$. This can be handled by shrinking the set T to a single node and solving the separation problem for the inequalities (2).

The second case considered in [19] is when $I_2 \neq \emptyset$. We remark that at least two sets V_j and V_k, $j \neq k$, should contain a terminal, that is $r(V_j) = r(V_k) = 2$, and thus $|I_2| \geq 2$. In consequence, in this case, the partition inequalities (7) can be written as

$$x(\delta(V_1, \ldots, V_p)) \geq p. \tag{8}$$

Kerivin and Mahjoub showed that the separation problem for the inequalities (8) is equivalent to minimizing, for every pair of terminals a and b, a particular submodular function. Because of the complexity of the submodular function minimization algorithms [17,28], this approach leads to an $O(n^{11})$ algorithm which cannot be considered practical.

In this paper we focus on the case $I_2 \neq \emptyset$ and reduce it to a sequence of submodular flow problems as it is shown in the next section

3. A submodular flow formulation

Here we deal with partitions separating two fixed terminals. Suppose each edge $e \in E$ has a weight $\bar{x}(e) \geq 0$. Let us consider two terminals t_1 and t_2 of T, $t_1 \neq t_2$. We are going to solve

$$\text{minimize } \bar{x}(\delta(V_1, \ldots, V_p)) - p \tag{9}$$

with the constraint that $t_1 \in V_1$ and $t_2 \in V_2$ say. This can be reduced to a submodular flow problem as described below.

For a node subset $W \subseteq V$, $W \neq \emptyset$, let

$$f_1(W) = \begin{cases} \bar{x}(\delta(W)) - 2 + M & \text{if } t_1 \in W, \\ \bar{x}(\delta(W)) - 2 & \text{if } t_1 \notin W, \end{cases}$$

and

$$f_2(W) = \begin{cases} \bar{x}(\delta(W)) - 2 + M & \text{if } t_2 \in W, \\ \bar{x}(\delta(W)) - 2 & \text{if } t_2 \notin W. \end{cases}$$

where M is a big value. And $f_1(\emptyset) = f_2(\emptyset) = 0$.

Lemma 1. *Both functions f_1 and f_2 are submodular on intersecting pairs.*

Proof. We only prove the result for the function f_1, the proof being similar for f_2. We must show that

$$f_1(A) + f_1(B) \geq f_1(A \cap B) + f_1(A \cup B) \tag{10}$$

for all intersecting pairs $A, B \subseteq V$. Let $A, B \subseteq V$ such that $A \cap B \neq \emptyset$, $A \setminus B \neq \emptyset$ and $B \setminus A \neq \emptyset$. We first notice that, since the vector $\bar{x}$ is nonnegative, we have

$$\bar{x}(\delta(A)) + \bar{x}(\delta(B)) \geq \bar{x}(\delta(A \cap B)) + \bar{x}(\delta(A \cup B)). \tag{11}$$

Moreover, the node t_1 belongs as many times to A and B as to $A \cap B$ and $A \cup B$. Thus, from (11), we can deduce the inequality (10). □

Let us associate a variable $y(u)$ to every node $u \in V$. From Lemma 1 and [8], it follows that the system

$$\begin{array}{ll} y(W) \leq f_1(W) & \text{for all } W \subseteq V, \\ y(W) \leq f_2(W) & \text{for all } W \subseteq V, \end{array}$$

is totally dual integral. Therefore, the dual of the following linear program

$$\text{maximize } y(V) \tag{12}$$

subject to

$$y(W) \leq f_1(W) \qquad \text{for all } W \subseteq V, \tag{13}$$

$$y(W) \leq f_2(W) \qquad \text{for all } W \subseteq V, \tag{14}$$

has an optimal solution that is integer valued. The dual program of (12)-(14) is as below

$$\text{minimize } \sum_{W \subseteq V} f_1(W)\alpha^1_W + \sum_{W \subseteq V} f_2(W)\alpha^2_W \tag{15}$$

subject to

$$\sum_{W \subseteq V : u \in W} \alpha^1_W + \sum_{W \subseteq V : u \in W} \alpha^2_W = 1 \qquad \text{for all } u \in V, \tag{16}$$

$$\alpha^1 \geq 0, \tag{17}$$

$$\alpha^2 \geq 0. \tag{18}$$

Lemma 2. *An integer optimal solution to the linear program (15)-(18) defines a partition of V which minimizes*

$$\bar{x}(\delta(V_1, \ldots, V_p)) - p \tag{19}$$

with the property that the nodes t_1 and t_2 appear in different sets of the partition.

Proof. First of all, we know that the system (13)-(14) is totally dual integral, and then the linear program (15)-(18) has an integer optimal solution. Let us denote by $(\bar{\alpha}^1, \bar{\alpha}^2)$ such a solution. Since the right-hand sides of the equations (16) are 1, and the dual variables are nonnegative, $(\bar{\alpha}^1, \bar{\alpha}^2)$ is clearly 0-1 valued.

Therefore, from the equations (16), any node u of V belongs exactly to one subset W of V with $\bar{\alpha}^1_W + \bar{\alpha}^2_W = 1$. Thus the family $\mathcal{F} = \{W : W \subset V,$ and either $\bar{\alpha}^1_W = 1$ or $\bar{\alpha}^2_W = 1\} = \{W_1, \ldots, W_q\}$ defines a partition of V.

Furthermore, because of the objective function (15), the nodes t_1 and t_2 belong to two different sets of the partition. In fact, this is the only manner to avoid having big value M in the objective function (15). The partition $\{W_1, \ldots, W_q\}$ gives

$$\sum_{W \subseteq V} f_1(W)\bar{\alpha}^1_W + \sum_{W \subseteq V} f_2(W)\bar{\alpha}^2_w = 2\bar{x}(\delta(W_1, \ldots, W_q)) - 2q,$$

and therefore, minimizes (19) with the constraint that the nodes t_1 and t_2 should appear in two different sets of the partition. □

If the value of (19) is greater or equal to 0, then it shows that all inequalities (8) induced by partitions of V with t_1 and t_2 in two different sets are satisfied by $\bar{x}$. If the value of this optimum is less than 0, then since the partition is obtained from an optimal solution of (15)-(18), we get the most violated inequality (8) induced by a partition of V with t_1 and t_2 in two different sets. This procedure has been described for two specific terminals t_1 and t_2 of T, now we can fix $t_1 \in T$ and try all $t_2 \in T \setminus \{t_1\}$.

In the next section, we discuss how to solve these submodular flow problems, that is, how to solve the linear programs (12)-(14) and (15)-(18).

4. The Fujishige-Zhang algorithm for the submodular intersection problem

In this section, we describe the algorithm of Fujishige and Zhang [9] for solving the linear programs (12)-(14) and (15)-(18). We consider throughout this section two fixed terminals t_1 and t_2, and their associated submodular functions on intersecting pairs f_1 and f_2 respectively.

4.1. Preliminaries

Given a ground set V, for a set-function $f : 2^V \to \mathbb{R} \cup \{\infty\}$, the following polyhedra are associated with f:

$$P(f) = \{y \in \mathbb{R}^V : y(A) \leq f(A) \text{ for all } A \subseteq V\},$$
$$B(f) = \{y \in \mathbb{R}^V : \ y(V) = f(V), \ y(A) \leq f(A) \text{ for all } A \subseteq V\}.$$

If f is submodular, then $P(f)$ is called the *submodular polyhedron* associated with f, and $B(f)$ is called the *base polyhedron* associated with f.

Let f be a set-function on V. The function $f' : 2^V \longrightarrow \mathbb{R} \cup \{\infty\}$ given by

$$f'(A) = \min\{\sum_i f(A_i) : \{A_i\} \text{ is a partition of } A, \ \emptyset \neq A_i \ \forall i\}$$

for $A \subseteq V$, $A \neq \emptyset$, $f'(\emptyset) = 0$, is called the *Dilworth truncation* of f. Notice that $f'(A) \leq f(A)$ for $\emptyset \neq A \subseteq V$. The following holds.

Theorem 1. *[22]. The Dilworth truncation f' of an intersecting submodular function f is fully submodular. Moreover, $P(f) = P(f')$.* □

Given the two fully submodular functions f_1' and f_2' on V, the *submodular intersection problem* is

$$\begin{aligned}&\text{maximize } y(V) &(20)\\&\text{subject to}\\&y \in P(f_1') \cap P(f_2'). &(21)\end{aligned}$$

It follows from [8] that the maximum in (20)-(21) is equal to

$$\text{minimize}\{f_1'(A) + f_2'(V \setminus A) : \ A \subseteq V\}. \quad (22)$$

Since functions f_1' and f_2' are the Dilworth truncations of f_1 and f_2 respectively, we have the following.

Lemma 3. *The minimum in (22) is exactly the minimum of the linear program (15)-(18).* □

To solve the problems (20)-(21) and (22), we use an algorithm given by Fujishige and Zhang [9]. To describe this algorithm, we need to introduce some notations. First, when we write f_i', we refer to one of both Dilworth truncations f_1' and f_2', and the subscript i may be either 1 or 2 for all the following notations.

A set $A \subseteq V$ with $y(A) = f_i'(A)$ is called *tight*$_i$ for y. Because of the submodularity of f_i', the union and the intersection of tight$_i$ sets are also tight$_i$.

For any $y \in P(f_i')$, let

$$\text{sat}_i(y) = \bigcup\{A \subseteq V : y(A) = f_i'(A)\}, \quad (23)$$

this the largest node subset of V tight$_i$ for y. The function $\text{sat}_i : P(f_i') \longrightarrow 2^V$ is called the *saturation function*.

For any $y \in P(f_i')$ and $u \in \text{sat}_i(y)$, let

$$\text{dep}_i(y,u) = \bigcap\{A \subseteq V : u \in A, \ y(A) = f_i'(A)\}, \quad (24)$$

this the smallest tight$_i$ set containing u. For any $y \in P(f_i')$ and $u \notin \text{sat}_i(y)$ we have $\text{dep}_i(y,e) = \emptyset$. The function $\text{dep}_i : P(f_i') \longrightarrow 2^V$ is called the *dependence function*.

For any $y \in P(f_i')$ and $u \in V$, the *saturation capacity* $\hat{c}_i(y,u)$ is defined by

$$\hat{c}_i(y,u) = \min\{f_i'(A) - y(A) : u \in A \subseteq V\}. \quad (25)$$

For any $y \in P(f_i')$ and $u, v \in V$, the *exchange capacity* $\tilde{c}_i(y,u,v)$ is defined by

$$\tilde{c}_i(y,u,v) = \min\{f_i'(A) - y(A) : u \in A \subseteq V, \ v \notin A\}. \quad (26)$$

Because of the definitions of the functions f_i' and f_i, we have the result below.

Lemma 4. *For any $y \in P(f_i')$ and $u, v \in V$, we have*

$$\begin{aligned}\hat{c}_i(y,u) &= \min\{f_i(A) - y(A) : u \in A \subseteq V\}, &(27)\\ \tilde{c}_i(y,u,v) &= \min\{f_i(A) - y(A) : u \in A \subseteq V, \ v \notin A\}. &(28)\end{aligned}$$

Proof. Given $y \in P(f_i')$ and $u \in V$, let $A^\star$ be a subset of V such that

$$\hat{c}_i(y,u) = f_i'(A^\star) - y(A^\star).$$

Since f_i' is the Dilworth truncation of f_i, there exists a partition $\{A_1^\star, \ldots, A_k^\star\}$ of $A^\star$ such that

$$\hat{c}_i(y,u) = \sum_{j=1}^{k} (f_i(A_j^\star) - y(A_j^\star)).$$

W.l.o.g., we may suppose $u \in A_1^\star$. The fact that $f_i(X) - y(X) \geq 0$ for all $X \subseteq V$ implies

$$\begin{aligned}\hat{c}_i(y,u) &\geq f_i(A_1^\star) - y(A_1^\star)\\ &\geq f_i'(A_1^\star) - y(A_1^\star) \geq \hat{c}_i(y,u).\end{aligned}$$

Therefore, we can deduce $\hat{c}_i(y,u) = \min\{f_i(A) - y(A) : u \in A \subseteq V\}$. The proof for the exchange capacity $\tilde{c}_i$ is similar. □

Now we show that computing the minimum in (27) and (28) reduces to a minimum cut problem. Similar constructions appear in [26], [27], [4] and [2]

Lemma 5. *The calculation of the minimum in* (27) *and* (28) *reduces to finding a minimum* st*-cut.*

Proof. Consider (28), and $i = 1$. Build a directed graph $D = (N, A)$, where $N = V \cup \{s,t\}$, and $A = \{(p,q),(q,p) \mid \text{ for } pq \in E\} \cup \{(s,p),(p,t) \mid \text{ for } p \in V\}$. Define capacities as follows:

$$\begin{aligned}
&c(s,p) = y(p), \quad c(p,t) = 0, \text{ if } y(p) > 0,\ p \in V \setminus \{u,v,t_1\},\\
&c(s,t_1) = y(t_1), \quad c(t_1,t) = M, \text{ if } y(t_1) > 0,\\
&c(p,t) = -y(p), \quad c(s,p) = 0, \text{ if } y(p) \leq 0,\ p \in V \setminus \{u,v,t_1\},\\
&c(t_1,t) = -y(t_1) + M, \quad c(s,t_1) = 0, \text{ if } y(t_1) \leq 0,\\
&c(s,u) = \infty, \quad c(u,t) = \max\{0, -y(u)\}, \quad c(s,v) = \max\{0, y(v)\}, \quad c(v,t) = \infty,\\
&c(p,q) = c(q,p) = \bar{x}(pq).
\end{aligned}$$

Let $\{s\} \cup S$ define a minimum st-cut. We should have $u \in S$ and $v \notin S$ because of the values of $c(s,u)$ and $c(v,t)$.

For any $S \subseteq V$ with $u \in S$ and $v \notin S$, let λ be the capacity of the cut defined by $\{s\} \cup S$. Then

$$x(\delta(S)) - y(S) = \begin{cases} \lambda - \sum\{y(p) \ : \ y(p) > 0\} & \text{if } t_1 \notin S,\\ \lambda - \sum\{y(p) \ : \ y(p) > 0\} - M & \text{if } t_1 \in S.\end{cases}$$

or

$$\lambda = \begin{cases} x(\delta(S)) - y(S) + \sum\{y(p) \ : \ y(p) > 0\} & \text{if } t_1 \notin S,\\ x(\delta(S)) - y(S) + \sum\{y(p) \ : \ y(p) > 0\} + M & \text{if } t_1 \in S.\end{cases}$$

Since $\sum\{y(p) \ : \ y(p) > 0\}$ is a constant, a minimum st-cut gives the minimum in (28). The other cases are similar □

4.2. The Algorithm

Fujishige and Zhang [9] extended the preflow-push algorithm of Goldberg and Tarjan [10] to the submodular intersection problem (20)-(21) as follows. Start with a pair $\beta = (y, z)$ fulfilling the conditions below

$$y \in B(f_1') \text{ and } z \in P(f_2'), \tag{29}$$
$$y \geq z. \tag{30}$$

This pair of vectors can be obtained as follows. Set $z(u) = -2$ for all u. Start with $y(u) = -2$ for all u, then increase $y(u)$ to $y(u) + \hat{c}_1(y, u)$ for each node u. The final vector y is in $B(f_1')$.

Use an auxiliary directed graph $\hat{G}_\beta = (\hat{V}, \hat{A}_\beta)$ defined as follows

$$\hat{V} = \{s^+, s^-\} \cup V,$$
$$\hat{A}_\beta = \hat{S}_\beta^+ \cup \hat{S}_\beta^- \cup \hat{A}_\beta^1 \cup \hat{A}_\beta^2,$$

where

$$\hat{S}_\beta^+ = \{(u, s^+) : u \in V\},$$
$$\hat{S}_\beta^- = \{(u, s^-) : u \in V \setminus \mathrm{sat}_2(z)\},$$
$$\hat{A}_\beta^1 = \big\{a : \partial^+ a = u,\ \partial^- a = v,\ u, v \in V,\ u \in \mathrm{dep}_1(y, v) \setminus \{v\}\big\},$$
$$\hat{A}_\beta^2 = \big\{a : \partial^+ a = u,\ \partial^- a = v,\ u, v \in V,\ v \in \mathrm{dep}_2(z, u) \setminus \{u\}\big\}.$$

Associated with each node $u \in V$, we define an *excess* $e(u) = y(u) - z(u)$. From condition (30), we notice that $e(u) \geq 0$ for $u \in V$. If $e(u) > 0$, then the node u is called *active*.

A function d from $\hat{V}$ to nonnegative integers is said to be a *valid labeling* for $\hat{G}_\beta$ if $d(s^+) = n + 2$, $d(s^-) = 0$ and $d(\partial^+ a) \leq d(\partial^- a) + 1$ for every arc $a \in \hat{A}_\beta$. For any valid labeling d, if $d(u) < n+2$, then $d(u)$ is a lower bound of the actual distance from the node u to s^-, where the length of each arc is equal to 1. If $d(u) \geq n + 2$, then $d(u) - (n + 2)$ is a lower bound of the actual distance from the node u to s^+ in $\hat{G}_\beta$ and s^- is not reachable from u in $\hat{G}_\beta$.

The initial valid labeling d is $d(s^+) = n + 2$, $d(s^-) = 0$ and $d(u) = 1$ for all $u \in V$. The algorithm then repetitively performs, in an order that will be mentioned later, the two basic operations "push" and "relabel" which are defined as follows.

Push(a): $a \in \hat{A}_\beta$;
Applicability: $\partial^+ a$ is active and $n + 2 \geq d(\partial^+ a) = d(\partial^- a) + 1$;
Action:

Case 1. If $a \in \hat{A}_\beta^1$ then put $y \leftarrow y + \alpha(\chi_v - \chi_u)$, where $u = \partial^+ a$, $v = \partial^- a$ and $\alpha = \min\{\tilde{c}_1(y, v, u), e(u)\}$.

Case 2. If $a \in \hat{A}_\beta^2$ then put $z \leftarrow z + \alpha(\chi_u - \chi_v)$, where $u = \partial^+ a$, $v = \partial^- a$ and $\alpha = \min\{\tilde{c}_2(y, u, v), e(u)\}$.

Case 3. If $a \in S^-_\beta$, then put $z \leftarrow z + \alpha\chi_u$, where $u = \partial^+ a$ and $\alpha = \min\{\tilde{c}_2(z,u), e(u)\}$.

Lemma 6. *[9]. Actions in all cases maintain the initial conditions (29) and (30) required for (y,z).* □

Relabel (u): $u \in V$;
Applicability: u is active and for any $a \in \hat{A}_\beta$ with $\partial^+ a = u$ we have $d(\partial^+ a) \leq d(\partial^- a)$;
Action: Put $d(u) \leftarrow \min\{d(v) + 1 : (u,v) \in \hat{A}_\beta\}$.

Lemma 7. *[9]. If $e(u) > 0$ and $d(u) \leq n+2$, then either a push for some $a \in \hat{A}_\beta$ with $\partial^+ a = u$ or a relabel of u is applicable.* □

Lemma 8. *[9]. The basic operations keep d a valid labeling.* □

Lemma 9. *[9]. For any $u \in U$, the distance label $d(u)$ never decreases by basic operations, and we have $d(u) \leq n + 3$.* □

Lemma 10. *[9]. Relabeling operations are carried out at most $n(n+2)$ times.* □

We give a detailed poof of the lemma below because it is needed to identify the optimal partition.

Lemma 11. *[9]. For a pair $\beta = (y,z)$ satisfying conditions (29) and (30), if there is no active node u in $\hat{G}_\beta$ with $d(u) \leq n + 2$, then z is a solution of (20)-(21).*

Proof. If there is no active node, then we have $y = z$ and z is a solution of (20)-(21). If there is an active node, let $U \subseteq V$ be the set of nodes in $\hat{G}_\beta$ which are reachable by directed paths from the active nodes. If $U = V$ and $V \setminus \mathrm{sat}_2(z) \neq \emptyset$, then there is an active node u such that s^- is reachable from u. This contradicts the fact that $d(u) = n + 3$. Therefore, if $U = V$, we have $\mathrm{sat}_2(z) = V$, which implies that z is a solution of (20)-(21).

Consider now the case when $U \neq V$. For $u \in V \setminus U$ we have $\mathrm{dep}_1(y,u) \subseteq V \setminus U$, otherwise there is an arc in $\hat{G}_\beta$ from a node in U to a node in $V \setminus U$. Thus

$$V \setminus U = \bigcup_{u \in V \setminus U} \mathrm{dep}_1(y,u).$$

Thus $V \setminus U$ is a union of tight$_1$ sets that by submodularity is also a tight$_1$ set. So $y(V \setminus U) = f'_1(V \setminus U)$.

Every node in U is in a tight$_2$ set, otherwise s^- would be reachable from an active node u which is impossible because $d(u) = n + 3$. Also for $u \in U$ we have $\mathrm{dep}_2(z,u) \subseteq U$, otherwise there is an arc in $\hat{G}_\beta$ from a node in U to a node in $V \setminus U$. Thus

$$U = \bigcup_{u \in U} \mathrm{dep}_2(z,u).$$

Therefore U is a union of tight$_2$ sets and is also tight$_2$ by submodularity. Thus $z(U) = f'_2(U)$.

We have $z(V) = z(U) + z(V \setminus U) = z(U) + y(V \setminus U)$ because every node in $V \setminus U$ is not active. Hence $z(V) = f'_2(U) + f'_1(V \setminus U)$, thus U gives the minimum of (22) and z is a solution of (20)-(21). □

In what follows we discuss the order in which the basic operations are performed. Let $\pi : \hat{V} \rightarrow \{1, 2, \ldots, n+2\}$ be a numbering of the nodes of $\hat{V}$. For any $u \in V$, there is an arc list $L_\beta(u)$ formed by the outgoing arc set $\{a : a \in \hat{A}_\beta,\ \partial^+ a = u\}$ arranged in the order of the increasing magnitude of the values of $\pi(\partial^- a)$. Each node has a *current arc* a in the list. Initially, the current arc of u is the first element of $L_\beta(u)$.

An active node v is selected such that

$$d(v) = \max\{d(w) : w \in V,\ d(w) \leq n+2,\ e(w) > 0\}.$$

Then, we have to check whether a push operation is applicable for the current arc a of $L_\beta(v)$. If the push operation is not applicable, then the next arc in $L_\beta(v)$, if any, becomes the current arc of v. If a push operation is applicable, then it is performed, and its result is either $e(v) = 0$, or $e(v) > 0$ and $a \notin \hat{A}_\beta$. In the first case, a new active node with the largest label is selected and the process is repeated. In the second case, the next arc in $L_\beta(v)$, if any, becomes the current arc of v. If the end of $L_\beta(v)$ is reached with $e(v) > 0$, then the first arc in the list becomes the current arc and a relabeling operation is carried out.

Lemma 12. *[9]. Throughout the algorithm the following property is maintained: For each $u \in V$, any arc a before the current arc in $L_\beta(u)$ satisfies $d(\partial^+ a) \leq d(\partial^- a)$.* □

A push on $(u, v) \in \hat{A}_\beta$ is called a *saturating push* if one of the following three conditions holds:
(a) The push is of Case 1 and $e(u) \geq \tilde{c}_1(y, v, u)$,
(b) The push is of Case 2 and $e(u) \geq \tilde{c}_2(z, u, v)$,
(c) The push is of Case 3 and $e(u) \geq \tilde{c}_2(z, u)$.

A push that is not saturating is called a *non-saturating push.*

Lemma 13. *[9]. The number of saturating push operations is at most $2n^2(n+2)$.* □

Lemma 14. *[9]. The number of nonsaturating pushes is at most $n^2(n+2)$.* □

Theorem 2. *[9]. The algorithm terminates after carrying out $O(n^2)$ relabeling operations and $O(n^3)$ push operations.* □

For our fully submodular functions f'_1 and f'_2 on V, we should keep for every node $u \in V$ an arc list $L_\beta(u)$ consisting of all arcs (u, v) with $v \in V \setminus \{u\} \cup \{s^+, s^-\}$. Then, when an arc becomes a candidate for a push operation, one should compute the exchange capacity or the saturation capacity associated with the current arc. Therefore, by Lemma 4 and Theorem 2, we deduce the complexity of the algorithm in our case.

Theorem 3. *Given the two fully submodular functions f'_1 and f'_2 on V, the Fujishige-Zhang algorithm requires $O(n^3)$ minimum st-cut computations.* □

4.3. Finding the partition

In order describe how to find an optimal partition we revisit the proof of Lemma 11. Let $(\bar{y}, \bar{z})$ be the pair of vectors produced by the algorithm. Let U be the set of nodes which are reachable by directed paths from the active nodes. First of all, we give the following lemma which goes through the different cases considered in the proof of Lemma 11, and then show that for our functions f'_1 and f'_2, only one of those cases can happen.

Lemma 15. *When the algorithm terminates, we have $\emptyset \neq U \subset V$.*

Proof. Suppose there is no active node. Then, we have $\bar{y} = \bar{z}$ and since $\bar{y} \in B(f'_1)$, we also have $\bar{y}(V) = f'_1(V)$. This implies that there exists a partition $\{V_1, \ldots, V_p\}$ of V with $\bar{y}(V_i) = f_1(V_i)$ for all $i = 1, \ldots, p$.

Consider the vector $(\bar{\alpha}^1, \bar{\alpha}^2)$ defined as $\bar{\alpha}^1_W = 1$ if $W = V_i$ for some i, $\bar{\alpha}^1_W = 0$ otherwise, $\bar{\alpha}^2 = 0$. This is an optimal solution of (15)-(18). From Lemmas 2 and 3, we know that t_1 and t_2 do not belong to the same set of the partition $\{V_1, \ldots, V_p\}$. W.l.o.g., we may assume that $t_1 \in V_1$ and $t_2 \in V_2$. By the definitions of f_1 and f_2, we obtain

$$\bar{z}(V_1) = \bar{y}(V_1) = \bar{x}(\delta(V_1)) - 1 + M > f_2(V_1) \geq f'_2(V_1),$$

a contradiction with $\bar{z} \in P(f'_2)$. Hence there exists at least one active node and then, $U \neq \emptyset$.

If $U = V$, then $\mathrm{sat}_2(\bar{z}) = V$ as it was shown in the proof of Lemma 11. Thus, we have $\bar{z}(V) = f'_2(V)$ and there exists a partition $\{U_1, \ldots, U_q\}$ of V with $\bar{z}(U_j) = f_2(U_j)$ for all $j = 1, \ldots, q$. As before, w.l.o.g. we assume $U_1 \cap \{t_1, t_2\} = \{t_2\}$. Since $(\bar{y}, \bar{z})$ fulfills conditions (29) and (30), by the definitions of f_1 and f_2, we obtain

$$\bar{x}(\delta(U_1)) - 1 + M = \bar{z}(U_1) \leq \bar{y}(U_1) \leq f'_1(U_1) \leq f_1(U_1) < \bar{x}(\delta(V_1)) - 1 + M,$$

a contradiction. Therefore, we deduce $U \subset V$. □

To obtain U one has to build the graph $\hat{G}_\beta$. For that one has to compute the saturation capacity of every node, and the exchange capacity for every arc. This requires $O(n^2)$ minimum *st*-cut computations. Then the set U is obtained by searching in $\hat{G}_\beta$ starting from the active nodes. This requires $O(n^2)$ time.

In what follows we show that the final partition is obtained from the sets $\mathrm{dep}_2(\bar{z}, u)$ for $u \in U$ and $\mathrm{dep}_1(\bar{y}, u)$ for $u \in V \setminus U$.

Lemma 16. *For $u \in U$ let $W = dep_2(\bar{z}, u)$, then $\bar{z}(W) = f_2(W)$.*

Proof. Because of the definition of f_2', there is a partition $\{W_i\}$ of W such that

$$0 = f_2'(W) - \bar{z}(W) = f_2(W_1) - \bar{z}(W_1) + \cdots + f_2(W_k) - \bar{z}(W_k).$$

Since $f_2(W_i) - \bar{z}(W_i) \geq 0$ for all i, we have $f_2(W_i) - \bar{z}(W_i) = 0$ for all i. We can assume that $u \in W_1$, but since W is the smallest tight$_2$ set containing u, we have $W_1 = W$. □

For each $u \in U$ we obtain $\text{dep}_2(\bar{z}, u)$ as

$$\text{dep}_2(\bar{z}, u) = \{v \;:\; \tilde{c}_2(\bar{z}, u, v) > 0\} \cup \{u\}.$$

This gives us a family of sets $\{U_i'\}$ whose union is U and such that $\bar{z}(U_i') = f_2(U_i')$ for all i. Finally some uncrossing should be done as follows. If $U_i' \cap U_j' \neq \emptyset$ and $U_i' \subseteq U_j'$ then only U_j' is kept. If $U_i' \cap U_j' \neq \emptyset$ and they are intersecting pairs, then $\bar{z}(U_i' \cup U_j') = f_2(U_i' \cup U_j')$, because f_2 is intersecting submodular. Thus we replace U_i' and U_j' by their union. This is repeated until no two sets intersect. This gives a partition $\{U_i\}$ of U such that $\bar{z}(U_i) = f_2(U_i)$ for all i.

Analogously for $u \in V \setminus U$ we obtain $\text{dep}_1(\bar{y}, u)$ as

$$\text{dep}_1(\bar{y}, u) = \{v \;:\; \tilde{c}_1(\bar{y}, u, v) > 0\} \cup \{u\}.$$

This gives us a family of sets $\{V_i'\}$ whose union is $V \setminus U$, and such that $\bar{y}(V_i') = f_1(V_i')$ for all i. Then we uncross them as above. This gives a partition $\{V_j\}$ of $V \setminus U$ such that $\bar{z}(V_j) = \bar{y}(V_j) = f_1(V_j)$, for all j.

The final partition is $\{\{U_i\}, \{V_j\}\}$.

5. Changing terminals

So far we have shown how to solve problem (9) for fixed terminals t_1 and t_2. Then one should repeat this for all choices of $t_2 \in T \setminus \{t_1\}$.

Suppose that t_2 is replaced by t_3, let f_3 be the function associated with t_3. Clearly the vector $\bar{y}$ will continue to satisfy $\bar{y} \in B(f_1')$. However the vector $\bar{z}$ might violate some constraint $z(S) \leq f_3'(S)$ with $t_2 \in S$. To fix that one should compute

$$\alpha = \min\{f_3(S) - \bar{z}(S) \mid t_2 \in S\},$$

and replace $\bar{z}(t_2)$ by $\min\{\bar{z}(t_2) + \alpha, \bar{y}(t_2)\}$. The value α is computed as in Lemma 5. This new pair $(\bar{y}, \bar{z})$ satisfies (29) and (30) and can be used to restart the algorithm.

Thus $O(n^4)$ is an upper bound on the number of minimum cuts that have to be found when all choices of $t_2 \in T \setminus \{t_1\}$ are made.

6. The 3-terminal case

In this section we show that the 3-terminal case is NP-hard. More precisely given a connected graph $G = (V, E)$ with positive edge-weights w and three distinct terminals $\{t_1, t_2, t_3\} \subset V$, consider

$$\text{minimize } w(\delta(V_1, \ldots, V_p)) - p \tag{31}$$

among all partitions $\{V_i\}$ of V with the constraint that $t_1 \in V_1$, $t_2 \in V_2$, $t_3 \in V_3$.

It was shown in [5] that the following problem is NP-hard.

$$\text{minimize } w(\delta(V_1, V_2, V_3)) \tag{32}$$

among all 3-partitions $\{V_1, V_2, V_3\}$ of V with the constraint that $t_1 \in V_1$, $t_2 \in V_2$, $t_3 \in V_3$.

In order to reduce problem (32) to (31), we assume that every edge has weight at least 1. Since the weights are positive we achieve this after dividing all weights by the minimum edge-weight. Then the result follows from the following lemma.

Lemma 17. *If all weights are at least 1, an optimal solution of* (31) *has* $p = 3$.

Proof. Consider a partition $\Phi = \{V_1, \cdots, V_p\}$ and a set V_i, $i \geq 4$. There is at least one edge between a node in V_i and a node in a set V_j, $j \neq i$. Since the weight of this edge is at least 1, when the sets V_i and V_j are combined into one we obtain a partition that is not worse than Φ. □

7. Final remarks

In this paper, we have studied the separation problem for the partition inequalities (7), that is, when we distinguish some terminal nodes. We have given an $O(n^7)$ algorithm which is based on Fujishige-Zhang algorithm for the submodular intersection problem. Nevertheless, our algorithm may lead to a time complexity that does not make it necessarily efficient in a branch-and-cut framework. Therefore, in this section, we give some remarks which may be considered in order to speed up the separation process in practice.

One should first solve the separation problem for inequalities (2) as shown in the next lemma.

Lemma 18. *Separating partition inequalities* (8) *violated by more than 1 reduces to separation of inequalities* (2).

Proof. Suppose that a violated inequality of type (2) is found. Let $\{V_1, \ldots, V_p\}$ be the associated partition. If all terminals are in one set, V_1 say, we have a violated inequality (7).

Otherwise we have

$$\bar{x}(\delta(V_1, \ldots, V_p)) - (p - 1) < 0.$$

And we deduce $\bar{x}(\delta(V_1, \ldots, V_p)) + 1 < p$. So this is an inequality (7) violated by more than 1. □

A second heuristic consists of finding a minimum cut $\delta(W)$ separating two terminals. Then solve the separation problem for inequalities (2) in the subgraphs induced by W and by $V \setminus W$. This is based on the lemma below.

Lemma 19. *Let $\{V_1, \ldots, V_p\}$ be a solution of (9), let G' be the subgraph induced by $V \setminus V_1$, then*

$$\bar{x}(\delta_{G'}(V_2, \ldots, V_p)) - (p-2) \leq 0.$$

Proof. If $\bar{x}(\delta_{G'}(V_2, \ldots, V_p)) - (p-2) > 0$ then $V_2, \ldots, V_p$ should be combined into one set to produce a better solution of (9). □

Acknowledgments

Part of this work was done while the second author was visiting the Watson Research Center of IBM in September 2003. The financial support of this institution is greatly appreciated.

References

1. M. Baïou, F. Barahona and A. R. Mahjoub, *Separation of partition inequalities*, Mathematics of Operations Research 25, (2000), pp. 243–254.
2. F. Barahona, *Separating from the dominant of the spanning tree polytope*, Operations Research Letters 12, (1992), pp. 201–203.
3. W. H. Cunningham, *Optimal attack and reinforcement of a network*, Journal of the Association for Computing Machinery 32, (1985), pp. 549–561.
4. W. H. Cunningham, *Minimum cuts, modular functions, and matroid polyhedra*, 15 (1985), pp. 205–215.
5. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, *The complexity of multiterminal cuts*, SIAM J. Comput., 23 (1994), pp. 864–894.
6. M. Didi Diha, H. Kerivin and A. R. Mahjoub, *(1,2)-survivable network design problem in series-parallel graphs*, in preparation, (2003).
7. J. Edmonds, *Submodular functions, matroid, and certain polyhedra*, in Combinatorial Structures and their Applications, R.K. Guy, E. Milner, and N. Sauer, eds., Gordon and Breach, New York, (1970), pp. 69–87.
8. J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, Annals of Discrete Mathematics 1, (1977), pp. 185–204.
9. S. Fujishige and X. Zhang, *New algorithm for the intersection problem of submodular system*, Japan Journal of Industrial and Applied Mathematics 9, (1992), pp. 369–382.
10. A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, Journal of the Association for Computing Machinery 35, (1988), pp. 921–940.
11. R. E. Gomory and T.C. Hu, *Multi-terminal network flows*, Journal of the Society for Industrial and Applied Mathematics 9, (1961), pp. 551–570.
12. M. Grötschel, L. Lovász and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer, (1988).
13. M. Grötschel and C. L. Monma, *Integer polyhedra arising from certain network design problems with connectivity constraints*, SIAM Journal of Discrete Mathematics 3, (1990), pp. 502–523.
14. M. Grötschel, C. L. Monma and M. Stoer, *Facets for polyhedra arising in the design of communication networks with low-connectivity constraints*, SIAM Journal on Optimization 2(3), (1992), pp. 474–504.
15. M. Grötschel, C. L. Monma and M. Stoer, *Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints*, Operations Research 40, (1992), pp. 309–330.

16. M. Grötschel, C. L. Monma and M. Stoer, *Design of survivable networks*, Handbook in Operations Research and Management Science 7, M. O. Ball et al., eds., North-Holland, (1995), pp. 617–671.
17. S. Iwata, L. Fleischer and S. Fujishige, *A combinatorial strongly polynomial algorithm for minimizing submodular functions*, Journal of the Association for Computing Machinery 48, (2001), pp. 761–777.
18. M. Jünger and W. R. Pulleyblank, *New primal and dual matching heuristics*, Report 91.105, Institut für Informatik, Universität zu Köln, (1991)
19. H. Kerivin and A. R. Mahjoub, *Separation of partition inequalities for the (1,2)-survivable network design problem*, Operations Research Letters 30, (2002), pp. 265–268.
20. H. Kerivin, C. Nocq and A. R. Mahjoub. *(1,2)-survivable networks: facets and branch-and-cut*, in Book: The Sharpest Cut, M. Grötschel, eds., MPS/SIAM on Optimization, pp. 139-173, to appear (Winter 2004).
21. J. B. Kruskal, *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proceedings of the American Mathematical Society 7, (1956), pp. 48–50.
22. L. Lovász, *Submodular functions and convexity*, in Mathematical Programming – The State of the Art, A. Bachem, M. Grötschel, B. Korte, eds., Springer, (1983), pp. 234–257.
23. A. R. Mahjoub, *Two edge connected spanning subgraphs and polyhedra*, Mathematical Programming 64, (1994), pp. 199-208.
24. C. L. Monma and D. F. Shallcross, *Methods for designing communication networks with certain two-connected survivability constraints*, Operations Research 37, (1989), pp. 531–541.
25. C. S. J. A. Nash-Williams, *Edge-disjoint spanning tree of finite graphs*, *J. London Math. Soc. 36*, (1961), pp. 445–450.
26. J. C. Picard and M. Queyranne, *Selected applications of minimum cuts in networks*, INFOR-Canada J. Oper. Res. Inform. Process., 20 (1982), pp. 394–422.
27. M. W. Padberg and A. Wolsey, *Trees and cuts*, Annals of Discrete Math., 17 (1983), pp. 511–517.
28. A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, Journal of Combinatorial Theory, Series B 80, (2000), pp. 346–355.
29. M. Stoer, *Design of survivable networks*, in Lecture Notes in Mathematics 1531, Springer-Verlag, (1992).
30. W. T. Tutte, *On the problem of decomposing a graph into n connected factors*, J. London Math. Soc. 36, (1961), pp. 221–230.