**RC25233 (W1111-058) November 8, 2011**
**Mathematics**

# IBM Research Report

## Partition Inequalities: Separation, Extensions and Network Design

**Mourad Baïou**
Université Blaise Pascal Clermont II
LIMOS CNRS UMR 6158
Complexe Scientifique des Cézeaux
63173 Aubiere Cedex, France

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

**A. Ridha Mahjoub**
Université Paris-Dauphine
LAMSADE CNRS UMR 7243
Pce Mal de Lattre de Tassigny
75775 Paris Cedex 16, France

# Partition Inequalities: Separation, Extensions and Network Design

Mourad Baïou[*], Francisco Barahona[†], A. Ridha Mahjoub[‡]

November 7, 2011

**Abstract**

Given a graph $G = (V, E)$ with nonnegative weights $x(e)$ for each edge $e$, a partition inequality is of the form $x\big(\delta(S_1, ..., S_p)\big) \geq ap + b$. Here $\delta(S_1, ..., S_p)$ denotes the multicut defined by a partition $S_1, ..., S_p$ of $V$. Partition inequalities arise as valid inequalities for optimization problems such as survivable network design problems, and play a central role in solving these problems using cutting planes. We attempt to survey some variants of these inequalities, examine different separation algorithms and discuss extensions and applications in network design and other domains.

*Keywords:* Partition inequality, separation, submodular function, *F*-partition, network design.

## 1 Introduction

Let $G = (V, E)$ be a graph with edge weights $x(e) \geq 0$ for all $e \in E$. Given a partition $S_1, ..., S_p$ of the node set $V$, we denote by $\delta(S_1, ..., S_p)$ the set of edges with endnodes in different sets of the partition. We use $\delta(S)$ instead of $\delta(S, V \setminus S)$ and we use $x(T)$ to denote $\sum_{e \in T} x(e)$.

Given $a$ and $b$, an inequality of the type

$$x\big(\delta(S_1, ..., S_p)\big) \geq ap + b \tag{1}$$

[*]Université Blaise Pascal Clermont II, LIMOS CNRS UMR 6158, Complexe scientifique des Céseaux, 63173 Aubière Cedex, France. Email: baiou@isima.fr

[†]IBM, Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA. Email: barahon@ibm.us.com

[‡]Université Paris-Dauphine, LAMSADE CNRS UMR 7243, Pce Mal de Lattre de Tassigny, 75775 Paris Cedex 16, France. Email: mahjoub@lamsade.dauphine.fr

is called a *partition inequality*. To motivate this, notice that the system of inequalities below defines a polyhedron whose extreme points are the incidence vectors of spanning trees, cf. [83, 73].

$$x\big(\delta(S_1,...,S_p)\big) \geq p-1, \quad \text{for all partitions } S_1,\ldots,S_p \text{ of } V,$$
$$x \geq 0.$$

As we shall see later, partition inequalities arise as valid inequalities or facets for optimization problems related to $k$-connectivity. In this paper we survey the *separation* problem: Given a vector $x$ find a violated inequality (1), if there is any. The separation problem is a key ingredient for being able to use these inequalities inside a cutting plane algorithm.

If $a \leq 0$ and there is a partition with $p > 2$ so that (1) is violated, we can collapse two sets, then the left hand side does not increase, and the right hand side does not decrease. So in this case one should only deal with $p = 2$ and the problem can be solved by finding a minimum cut.

We have to treat with the case $a > 0$, and without loss of generality we can assume that $a = 1$. Then we study the problem

$$\text{minimize } x\big(\delta(S_1,...,S_p)\big) - p, \tag{2}$$

where the minimization is among all partitions of $V$. This is the subject of Section 2.

Consider first the case when $b \leq -1$. If the minimum in (2) is given by the trivial partition ($p = 1$), then there is no violated inequality. If the minimum is given by a different partition we just have to compare this value with $b$. For the case when $b > -1$, we could have that the minimum in (2) is given by the trivial partition, but it could exist a violated partition inequality with $p \geq 2$. Thus in this case we study (2) with the constraint "$p \geq 2$." This case is somewhat harder, and it is treated in Section 3.

In some situations we might have some special set of nodes $T = \{t_1,\ldots,t_r\}$ called *terminals*. Then we might have the additional condition that $T$ should not be included in a set $S_i$, i.e., at least two terminals should be in different sets of the partition. This case is treated in Section 4.

In Section 5 we discuss extensions like the strength of a network, the principal sequence of partitions of a graph, network reinforcement, packing spanning trees, increasing the weight of minimum spanning trees and the Potts' model in Statistical Physics.

As we shall see, the main ingredients used are submodular functions, polymatroids, submodular flows and minimum cuts. We conclude with some notation and definitions. Given a ground set $S$, a set-function $f: 2^S \longrightarrow \mathbb{R} \cup \{\infty\}$ is called *fully submodular* if

$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \tag{3}$$

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 (3) is required only for intersecting pairs.

Sections 6, 7 and 8 are devoted to some applications of partition inequalities to network design. In particular we consider optimization problems related to survivability in telecommunication networks. Partition inequalities have shown to be very efficient for solving these problems using cutting plane based algorithms. In Section 6 we present some valid inequalities. In Section 7 we discuss the concept of the critical extreme points of the 2-edge connected subgraph polytope and show how the so-called *F*-partition inequalities may be used to cut those extreme points. In Section 8 we discuss applications to survivable networks when bound constraints are considered.

Throughout this paper we deal with a graph $G = (V, E)$, we use $n$ to denote $|V|$ and $m$ to denote $|E|$.

# 2 The case $b \leq -1$

Cunningham [19] studied the *attack problem*, defined as follows. Given a graph $G = (V, E)$ with edge weights $w(e) \geq 0$, for all $e \in E$, and a number $\lambda > 0$; find and edge set $A$ that minimizes

$$w(A) - \lambda\big(c(E \setminus A) - 1\big),$$

where $c(E \setminus A)$ is the number of connected components of $G$ after deleting the edges in $A$. One can visualize $w(e)$ as the effort required by an attacker to destroy the edge $e$, and $\lambda$ as the benefit to the attacker for each additional component created. This reduces to

$$\text{minimize } w\big(\delta(S_1, ..., S_p)\big) - \lambda(p - 1),$$

among all partitions of $V$. Cunningham gave an algorithm that requires solving $m$ minimum cut problems. Later algorithms that require $n$ minimum cut problems were given in [75, 7, 77]. Below we present the algorithm of [7].

## 2.1 The attack problem

We shall see that this problem reduces to optimizing a linear function over an *extended polymatroid*. These concepts are discussed in [45] for instance. This problem can be solved with the greedy algorithm used by Edmonds [30]. At each iteration, finding an inequality that becomes tight reduces to finding a minimum cut in a network.

Given a graph $G = (V, E)$ the spanning tree polytope $T(G)$ is the convex hull of incidence vectors of spanning trees of $G$, its dominant is the polyhedron $P(G) = T(G) + \mathbb{R}^E_+$ obtained by adding the nonnegative orthant. It has been proved in [83, 73] that $P(G)$ is defined by

$$x(\delta(V_1, ..., V_p)) \geq p - 1, \text{ for every partition of } V, \tag{4}$$

$$x \geq 0. \tag{5}$$

Jünger and Pulleyblank [57] have given an extended formulation for $P(G)$ as follows. Associate the variables $x$ with the edges and the variables $y$ with the nodes. The system below defines a polyhedron whose projection onto the variables $x$ is $P(G)$. This will be proved at the end of this section. The node $r$ is an arbitrary element of $V$.

$$x(\delta(S)) + y(S) \geq 2, \text{ if } r \notin S, S \subset V, \tag{6}$$
$$x(\delta(S)) + y(S) \geq 0, \text{ if } r \in S, S \subset V, \tag{7}$$
$$y(V) = 0, \tag{8}$$
$$x \geq 0. \tag{9}$$

So given a vector $\bar{x} \geq 0$, to check if inequalities (4) are satisfied, we can try to find a vector $\bar{y}$ such that $(\bar{x}, \bar{y})$ satisfies (6)-(9), or prove that $\bar{y}$ does not exist.

Let

$$f(S) = \begin{cases} 2 - \bar{x}\,(\delta(S)), & \text{if } r \notin S, \\ -\bar{x}\,(\delta(S)), & \text{if } r \in S, \end{cases}$$

for $\emptyset \neq S \subseteq V$. The function $-f$ is submodular on intersecting pairs.

We are going to solve

$$\begin{array}{l} \text{minimize } y(V) \\ \text{subject to} \\ y(S) \geq f(S), \text{ for } S \subseteq V. \end{array} \tag{10}$$

Edmonds [30] showed that the greedy algorithm solves this linear program. This algorithm, which we present below, produces also an optimal solution of the dual problem. We shall see that this gives a most violated partition inequality, if there is any. The dual problem is

$$\begin{array}{l} \text{maximize } \sum z_S\, f(S) \\ \text{subject to} \\ \sum\{z_S | u \in S\} = 1, \text{ for all } u \in V, \\ z \geq 0. \end{array} \tag{11}$$

Given a vector $y$ satisfying (10), a set $S$ is called *tight* if $y(S) = f(S)$. The function $y(\cdot) - f(\cdot)$ is nonnegative and submodular on intersecting pairs. So if $S$ and $T$ are tight, and $S \cap T \neq \emptyset$, then $S \cap T$ and $S \cup T$ are also tight.

We start with $y(v_i) = 2\ \forall i$, and decrease the value of each $y(v_i)$ until a set becomes tight. We denote by $\mathcal{F}$ the family of tight sets with a positive dual variable. If we try to add $S$ to $\mathcal{F}$ and there is a set $T \in \mathcal{F}$ with $S \cap T \neq \emptyset$, then we replace $S$ and $T$ by $S \cup T$. This is also tight.

Let $V = \{v_1, ..., v_n\}$, the algorithm is below.

**Algorithm A**

**Step 0**. Set $\bar{y}(v_i) \leftarrow 2$ for $i = 1, ..., n$; $k \leftarrow 1$; $\mathcal{F} \leftarrow \emptyset$.

**Step 1**. If $v_k$ belongs to a set in $\mathcal{F}$ go to Step 3, otherwise
set $\alpha \leftarrow f(\bar{S}) - \bar{y}(\bar{S}) = \max\{f(S) - \bar{y}(S) \,|\, v_k \in S\}$,
$\bar{y}(v_k) \leftarrow \bar{y}(v_k) + \alpha$,
$\mathcal{F} \leftarrow \mathcal{F} \cup \{\bar{S}\}$.

**Step 2**. While there are two sets $S$ and $T$ in $\mathcal{F}$ with $S \cap T \neq \emptyset$ do
$\mathcal{F} \leftarrow (\mathcal{F} \backslash \{S, T\}) \cup \{T \cup S\}$.

**Step 3**. Set $k \leftarrow k+1$, if $k \leq n$ go to Step 1, otherwise stop.

The vector $\bar{y}$ is built so it satisfies (10), the family $\mathcal{F}$ defines a partition of $V$ and $\bar{y}(S) = f(S)$ for every $S \in \mathcal{F}$. We set $\bar{z}_S = 1$, if $S \in \mathcal{F}$, and $\bar{z}_S = 0$ otherwise. We have

$$\bar{y}(V) = \sum\{\bar{y}\,(S) \,|\, S \in \mathcal{F}\} = \sum\{f(S) \,|\, S \in \mathcal{F}\} = \sum\{f(S)\,\bar{z}_S \,|\, S \subseteq V\}.$$

This proves that $\bar{y}$ and $\bar{z}$ are optimal solutions.

If the value of the optimum is 0 then $(\bar{x}, \bar{y})$ satisfies (6), (7), (8). In this case we can pick any partition of $V$ into $V_1, ..., V_p$, add the inequalities in (6), (7) associated with the sets $\{V_i\}$, and $-y(V) \geq 0$. We obtain a partition inequality. This shows that $\bar{x}$ satisfies all the partition inequalities.

Now assume that the value of the optimum is greater than 0. Let $\bar{z}$ be a 0-1 vector that satisfies the equations of (11), the family $\mathcal{G} = \{S \,|\, \bar{z}_S = 1\} = \{S_1, ..., S_p\}$ gives a partition of the set $V$ and

$$\sum f(S)\,\bar{z}_S = 2(p-1) - 2\,\bar{x}\,(\delta(S_1, ..., S_p)),$$

so

$$\frac{1}{2}\sum f(S)\,\bar{z}_S = (p-1) - \,\bar{x}\,(\delta(S_1, ..., S_p)).$$

If $\bar{z}$ is an optimum of (11), then it gives a most violated partition inequality. Because of this we have a solution of the separation problem not only for $b = -1$, but also for any $b \leq -1$.

This procedure shows that (11) has an optimal integer solution, this is known as the *total dual integrality* of (10). It also shows that $\bar{x}$ satisfies (4)-(5) if and only if there is a vector $\bar{y}$ such that $(\bar{x}, \bar{y})$ satisfies (6)-(9), so projecting the variables $y$ in (6)-(9), gives (4)-(5).

## 2.2 Finding tight sets

It remains to show how to compute the number $\alpha$ in Step 1. Construct a directed graph $D = (N, A)$, where $N = V \cup \{s, t\}$ and

$$A = \{(i,j), (j,i) \,|\, ij \in E\} \cup \{(s,i), (i,t) \,|\, i \in V\}.$$

Define

$$\begin{aligned} &\eta(i) = \bar{y}(i), \text{ for } i \in V, i \neq r, \\ &\eta(r) = \bar{y}(r) + 2, \end{aligned}$$

define capacities

$$\begin{aligned} &c(s,i) = -\eta(i),\; c(i,t) = 0,\; \text{if } \eta(i) < 0,\; i \neq v_k,\; i \in V, \\ &c(i,t) = \eta(i),\; c(s,i) = 0,\; \text{if } \eta(i) \geq 0,\; i \neq v_k,\; i \in V, \\ &c(s,v_k) = \infty,\; c(v_k,t) = \eta(v_k), \\ &c(i,j) = c(j,i) = \bar{x}(i,j),\; \text{for } ij \in E. \end{aligned}$$

**Lemma 1** *Suppose that* $\{s\} \cup T$ *induces a cut separating* $s$ *from* $t$ *that has capacity* $\lambda$*. Then*

$$\bar{y}(T) + \bar{x}(\delta(T)) = \begin{cases} \lambda + \sum\{\eta(v) \,|\, \eta(v) < 0\} - 2, & \textit{if } r \in T \\ \lambda + \sum\{\eta(v) \,|\, \eta(v) < 0\} & \textit{if } r \notin T. \end{cases}$$

**Proof.** Suppose that $r \in T$. Then

$$\lambda = \sum\{-\eta(i)|i \notin T, \eta(i) < 0\} + \bar{x}(\delta(T)) + \sum\{\eta(i)|i \in T, \eta(i) \geq 0\},$$

and

$$\begin{aligned} &\lambda + \sum\{\eta(v) \,|\, \eta(v) < 0\} - 2 = \\ &\sum\{\eta(i)|i \in T, \eta(i) < 0\} + \bar{x}(\delta(T)) + \sum\{\eta(i)|i \in T, \eta(i) \geq 0\} - 2 = \\ &\bar{y}(T) + \bar{x}(\delta(T)). \end{aligned}$$

The case $r \notin T$ is analogous. □

Therefore if $\beta$ is the minimum capacity of a cut separating $s$ from $t$, then the value $\alpha$ is

$$2 - \beta - \sum\{\eta(v) \,|\, \eta(v) < 0\}.$$

Suppose now that we have solved problem (10), and that we add a new vertex to the graph. We are going to show that resolving (10) takes just one min-cut calculation. This will be used in the next section.

**Lemma 2** *After solving problem* (10)*, if we add a new vertex, it will take one minimum cut calculation to resolve* (10)*.*

**Proof.** Suppose that $\bar{y}$ is the solution of (10), for the graph $G = (V, E)$. Suppose that we add the vertex $w$. Define

$$\bar{\bar{y}}(v) = \bar{y}(v) - \bar{x}(wv),$$

for all $v \in V$, and $\bar{\bar{y}}(w) = 2$. It is easy to see that $\bar{\bar{y}}$ satisfies the inequalities of (10), and every set that was tight before will remain tight. Thus the only component that can be modified by the greedy algorithm is $y(w)$, this takes one minimum cut problem. □

This Lemma shows that we can solve (10) adding the nodes one by one. Then at iteration $i$ one has to solve a minimum cut problem with $i + 2$ nodes.

**Lemma 3** *If the set* $\bar{S}$ *in Step 1 is of maximum cardinality, then Step 2 is not needed.*

**Proof.** If $\bar{S}$ is of maximum cardinality then it is a maximal tight set, and no uncrossing is needed. □

The preflow-push algorithm of [44] produces a minimum cut so that the source side has maximum cardinality, so it produces the set needed in Lemma 3. One should also notice that because of this noncrossing property, each tight set in $\mathcal{F}$ can be shrunk to a single node and the minimum cut problem is solved in a smaller graph. The algorithm given in [77] has similar properties.

This concludes the treatment of the case when $b \leq -1$. We have seen that it reduces to $n$ minimum cut problems.

# 3 The case $b > -1$

As we mentioned in the Introduction, the case when $b > -1$ will be treated as

$$\text{minimize } x(\delta(S_1, ..., S_p)) - p, \tag{12}$$

with $p \geq 2$. We follow the approach presented in [5].

This is equivalent to minimize

$$g(S) = x(\delta(S)) - 1 + \min \Big\{ x(\delta_S(T_1, ..., T_k)) - k \Big\}, \tag{13}$$

where $\emptyset \neq S \subset V$ and $\{T_i\}$ is a partition of $S$. Notice that $\{T_i\}$ could be the trivial partition, i.e., $k = 1$. In this section we use $\delta_S(T_1, ..., T_k)$ to denote the set of edges with endnodes in different sets $T_i$. The resulting partition is $\{\bar{S}, T_1, ..., T_k\}$.

First we have to see that $g$ is submodular. Consider

$$f(S) = x(\delta(S)) - 2,$$

for $S \subseteq V$. The function $f$ is submodular. 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 4** *[63]. The Dilworth truncation of a submodular function on intersecting pairs is fully submodular.*

We have that

$$g(S) = \frac{1}{2}\Big( x(\delta(S)) - 2 + f'(S) \Big),$$

therefore $g$ is submodular.

Queyranne [78] gave an algorithm to minimize a symmetric submodular function $h$ that takes $O(n^3)$ evaluations of the function. Symmetric means that $h(S) = h(\bar{S})$ for all $S \subseteq V$. Since $g$ is not symmetric, we define $g'(S) = f'(S) + f'(\bar{S})$ and look for the minimum of $g'(S)$, for $\emptyset \neq S \subset V$. It is clear that $g'$ is symmetric, and submodular because it is the sum of submodular functions.

## 3.1 Queyranne's algorithm

An algorithm to minimize a symmetric submodular function $h$ was given in [78], it generalizes the minimum cut algorithm of Nagamochi and Ibaraki [69] as simplified by Stoer and Wagner [82] and Frank [37]. The algorithm is below, we use $S + u$ to denote $S \cup \{u\}$.

**Algorithm B**

**Step 0.** Start with $W_0 = \emptyset$, $i = 0$.

**Step 1.** For all $u \notin W_i$ set $k(u) = h(W_i + u) - h(u)$. Let $k(u_{i+1}) = \min\{k(u)\}$.

**Step 2.** Set $W_{i+1} \leftarrow W_i + u_{i+1}$, set $i \leftarrow i + 1$. If $i = n$ stop, otherwise go to Step 1.

In [78] it was proved that

$$h(u_n) = \min\{h(S) \mid S \text{ separates } u_n \text{ and } u_{n-1}\}.$$

The next step is to identify $u_n$ and $u_{n-1}$, apply Algorithm B, and continue until we are left with two elements.

If we apply this algorithm with a submodular function $h$ that is not symmetric, we obtain the minimum of $h(S) + h(\bar{S})$, cf. [78]. In our case we just have to use the function $f'$ defined above.

Each application of Algorithm B requires $O(n^2)$ evaluations of $h$. Since Algorithm B is used $n - 1$ times, we need $O(n^3)$ evaluations of $h$. In our case, one evaluation of the function $f'$ with the algorithm of Section 2 takes $O(n)$ min-cut problems, thus the straightforward implementation of this method requires $O(n^4)$ min-cut problems. However each evaluation in Step 1 is of the type $f'(W_i + u)$ where $f'(W_i)$ is already known. We have seen at the end of Section 2 that this takes only one min-cut calculation. Thus the entire algorithm requires $O(n^3)$ min-cut problems.

# 4 Partition inequalities with terminals

Given a set of terminals $T = \{t_1, \ldots, t_k\}$, we need partitions so that the set of terminals intersects at least two sets of the partition. In [59] it was shown that this reduces to minimizing a submodular function, later in [11] a reduction to submodular flows was given. We present the latter approach here.

We fix two terminals and look for partitions separating them. 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

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

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 5** *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

$$f_1(A) + f_1(B) \geq f_1(A \cap B) + f_1(A \cup B) \tag{14}$$

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

$$\bar{x}(\delta(A)) + \bar{x}(\delta(B)) \geq \bar{x}(\delta(A \cap B)) + \bar{x}(\delta(A \cup B)). \tag{15}$$

Moreover, the node $t_1$ belongs as many times to $A$ and $B$ as to $A \cap B$ and $A \cup B$. Thus, from (15), we can deduce the inequality (14). $\square$

Let us associate a variable $y(u)$ to every node $u \in V$. From Lemma 5 and [31], 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

$$\text{maximize } y(V) \tag{16}$$

subject to

$$y(W) \leq f_1(W) \qquad \text{for all } W \subseteq V, \tag{17}$$

$$y(W) \leq f_2(W) \qquad \text{for all } W \subseteq V, \tag{18}$$

has an optimal solution that is integer valued. The dual program of (16)-(18) is the following:

$$\text{minimize } \sum_{W \subseteq V} f_1(W)\alpha^1_W + \sum_{W \subseteq V} f_2(W)\alpha^2_W \tag{19}$$

subject to

$$\sum_{W \subseteq V : u \in W} \alpha^1_W + \sum_{W \subseteq V : u \in W} \alpha^2_W = 1 \qquad \text{for all } u \in V, \tag{20}$$

$$\alpha^1 \geq 0, \tag{21}$$

$$\alpha^2 \geq 0. \tag{22}$$

**Lemma 6** *An integer optimal solution to the linear program (19)-(22) defines a partition of $V$ which minimizes*

$$\bar{x}(\delta(V_1, \ldots, V_p)) - p \tag{23}$$

*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 (17)-(18) is totally dual integral, and then the linear program (19)-(22) 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 (20) are 1, and the dual variables are nonnegative, $(\bar{\alpha}^1, \bar{\alpha}^2)$ is clearly 0-1 valued.

Therefore, from the equations (20), any node $u$ of $V$ belongs exactly to one subset $W$ of $V$ with $\bar{\alpha}_W^1 + \bar{\alpha}_W^2 = 1$. Thus the family $\mathcal{F} = \{W : W \subset V$, and either $\bar{\alpha}_W^1 = 1$ or $\bar{\alpha}_W^2 = 1\} = \{W_1, \ldots, W_q\}$ defines a partition of $V$.

Furthermore, because of the objective function (19), 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 (19). The partition $\{W_1, \ldots, W_q\}$ gives

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

and therefore, minimizes (23) with the constraint that the nodes $t_1$ and $t_2$ should appear in two different sets of the partition. □

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 $t_2 \in T \setminus \{t_1\}$. These submodular flow problems can be solved with the algorithm of Fujishige and Zhang [40], a detailed description of this is in [11]. One application of this requires $O(n^3)$ minimum cut problems, so to treat all terminals we need $O(n^4)$ minimum cut problems.

# 5 Extensions

Here we present several extensions of the problem studied in Section 2.

## 5.1 The strength of a network

Consider a graph $G = (V, E)$, it has been proved in [83] and [73] that the maximum number of disjoint spanning trees in $G$ is

$$\sigma = \min \left\lfloor \frac{|\delta(S_1, \ldots, S_p)|}{p-1} \right\rfloor, \tag{24}$$

where the minimum in (24) is taken among all partitions $\{S_1, \ldots, S_p\}$ of $V$ with $p \geq 2$. The number $\sigma$ has been proposed as a measure of the invulnerability of a network in [49].

In general suppose that each edge $e$ has a *strength* $s(e) \geq 0$. The *strength* of this network is

$$\sigma(G, s) = \min \frac{s\big(\delta(S_1, \ldots, S_p)\big)}{p-1} \tag{25}$$

where the minimum in (25) is taken among all partitions $\{S_1, \ldots, S_p\}$ of $V$ with $p \geq 2$.

Algorithms for computing $\sigma(G, s)$ have been given in [19], [50], [41] and [15]. In this last reference it is shown that it can be computed in the same asymptotic complexity as $n$ applications of the minimum cut algorithm of [44]; this is based on the parametric minimum cut algorithm of [43]. Now we describe this approach. One has to apply Dinkelbach's method [29] as follows.

**Algorithm C**

**Step 0.** Pick any partition $\{S_1, \ldots, S_l\}$ of $V$, $l \geq 2$. Set $\lambda = s\big(\delta(S_1, \ldots, S_l)\big)/(l-1)$.

**Step 1.** Solve

$$\text{minimize } s\big(\delta(T_1, \ldots, T_p)\big) - \lambda(p-1)$$

**Step 2.** If the minimum above is zero, stop. Otherwise let $\{U_1, \ldots, U_r\}$ be a solution. Set $\lambda = s\big(\delta(U_1, \ldots, U_r)\big)/(r-1)$, and go to Step 1.

Cunningham [19] showed that the value $r$ in Step 2 decreases at every iteration, so this algorithm takes at most $|V| - 2$ iterations. The minimization in Step 1 can be done by solving (10), where the right hand side of (6) should be $2\lambda$ instead of 2. Since the value of $\lambda$ decreases at every iteration, an optimal solution of (10) is feasible for the next value of $\lambda$.

Now consider the minimum cut problem that one has to solve for each node as in Subsection 2.2. For the source $s$, capacities $c(s,i)$ do not decrease from one iteration to the next. For the sink $t$, capacities $c(i,t)$ do not increase. The capacities of all other arcs remain the same. These properties permit the use of the parametric minimum cut algorithm of Gallo et al. [43]. For every node $v$ one has to solve a sequence of minimum cut problems, (one for each value of $\lambda$). This sequence can be solved with the same asymptotic complexity as one application of the preflow-push algorithm of [44]. Since we have such a sequence for each node, the complexity of this procedure is $O(n^4)$. Notice that one has to keep the data for each sequence, so the storage required is $O(nm)$.

## 5.2 Principal sequence of partitions of a graph

Let us denote by $P(\lambda)$ the following parametric problem. For a graph $G = (V, E)$ with edge weights $w(e) \geq 0$ for each $e \in E$, and a parameter $\lambda \geq 0$, solve

$$\text{minimize } w(\delta(S_1, ..., S_p)) - \lambda p, \tag{26}$$

where the minimization is over all partitions of $V$.

If $\Pi = \{S_1, \ldots, S_p\}$ is a partition of $V$, the sets $S_i$ are called the *blocks* of the partition. We have a partial order "$\succeq$" on the set of partitions, where $\Pi_1 \succeq \Pi_2$ if and only if each block of $\Pi_2$ is contained in a block of $\Pi_1$.

The set of partitions that are either maximal or minimal optimal solutions of $P(\lambda)$ for at least one $\lambda$ can be arranged in a decreasing sequence $\Pi_1, \ldots, \Pi_r$ called the *principal sequence of partitions* of the graph $G$. Each successive pair $\Pi_i, \Pi_{i+1}$ in the sequence consists of the maximal and minimal optimal partition for some (unique) value of $\lambda$. The resulting increasing sequence of $\lambda$'s is called the sequence of critical values of the principal sequence of partitions of $G$. Notice that the first partition consists of just the node-set $V$, and the last partition consists of all singletons.

Several algorithms have been proposed for this, see [54, 70, 75, 76, 62]. Here we discuss the approach of Kolmogorov [62].

Define

$$f^\lambda(S) = w(\delta(S)) - 2\lambda,$$

for $\emptyset \neq S \subseteq V$. The function $f$ is submodular for intersecting pairs. Then the linear program below can be solved with the greedy algorithm [30].

$$\begin{aligned}&\text{maximize } y(V)\\&y(S) \leq f^{\lambda}(S), \text{ for all } S, \emptyset \neq S \subseteq V.\end{aligned}$$

As in Section 2, the dual solution gives a partition that solves (26). At each iteration of the greedy algorithm one has to increase the value of $y(v)$ for some node $v$. The amount that one can increase is obtained by solving a minimum cut problem. So the greedy algorithm requires solving $n$ minimum cut problems.

Kolmogorov [62] showed how to modify this approach so that one can use the parametric minimum cut algorithm of Gallo et al [43], and solve $P(\lambda)$ for all $\lambda \geq 0$, in the same asymptotic complexity as $n$ applications of the preflow-push algorithm of [44].

One can use $P(\lambda)$ and Lagrangian relaxation to derive a lower bound for the $k$-cut problem, see [9], [79]. Also approximation algorithms based on $P(\lambda)$, for the $k$-cut problem and min-$k$-overlap have been given in [72].

The minimum critical value also gives the strength of a network, cf. [39]. To see this, notice that one can apply Algorithm C from Subsection 5.1, using the solutions of $P(\lambda)$. Then one can see that the value of $\lambda$ given by Algorithm C is exactly the smallest critical value.

For applications of these concepts to Electrical Network theory see [55, 71].

### 5.3 Network reinforcement

Now suppose that there is a per-unit cost $c(e)$ of increasing the strength of each edge $e$, and a number $\sigma_0$. The *reinforcement problem* consists of finding a minimum cost way to increase edge strengths so that the resulting network has strength at least $\sigma_0$. An algorithm for this was given in [19], it requires solving $2m$ minimum cut problems. Later an algorithm was given in [41] that requires $n$ iterations, each of them consisting of three steps. The first step uses the parametric network flow algorithm of [1], the second step requires an adaptation of the Hao-Orlin minimum cut algorithm [51], the third step uses the original Hao-Orlin algorithm. The algorithm of [41] requires $O(m)$ space.

In [10] an algorithm was given that has the same asymptotic complexity as $n$ applications of the minimum cut algorithm of [44]. This set of minimum cut problems has to be kept in memory simultaneously, so the space required is $O(nm)$. This algorithm has the same asymptotic complexity as the one of [41] although it is quite different.

Now we describe the basics of this approach. Consider a slightly different question, suppose that each edge $e$ has a nonnegative per-unit cost $d(e)$ and a nonnegative integer capacity $u(e)$, that gives the maximum number of copies allowed of edge $e$. For a nonnegative number $k$, consider the problem of choosing a minimum cost spanning subgraph of strength $k$. This can be modeled as the linear program below.

$$\min dx \tag{27}$$

subject to

$$x\big(\delta(S_1, \ldots, S_p)\big) \geq k(p-1), \text{ for all partitions } \{S_1, \ldots, S_p\} \text{ of } V, \tag{28}$$

$$0 \leq x(e) \leq u(e). \tag{29}$$

We can reduce the reinforcement problem to this problem by allowing parallel edges and giving the cost zero to the already existing edges. For the case when $u(e) = 1$ for all $e \in E$, an $O(m \log m + k^2 n^2)$ algorithm has been given in [80].

As seen in Section 2, this is equivalent to the linear program

$$
\begin{aligned}
&\min \sum d(e)x(e) \\
&x(\delta(S)) + y(S) \geq \begin{cases} 2k & \text{if } r \notin S,\ S \subset V, \\ 0 & \text{if } r \in S,\ S \subset V, \end{cases} \\
&y(V) = 0, \\
&-x \geq -u, \\
&x \geq 0.
\end{aligned}
$$

A combinatorial algorithm was given in [10]. Its complexity is dominated by the complexity of $n$ applications of the preflow-push algorithm of [44]. Related reinforcement questions have been studied in [76].

## 5.4 Packing spanning trees

We have seen that the minimum in (24) gives the maximum number of disjoint spanning trees in a graph. There are several algorithms to solve (24), but they do not give a maximum set of disjoint spanning trees. Given a graph $G = (V, E)$, with edge capacities $w(e) \geq 0$ for all $e \in E$, consider the following problem:

$$
\begin{aligned}
&\text{maximize } \sum \{\lambda_T \,:\, T \text{ is a spanning tree}\} \\
&\text{subject to} \\
&\sum \{\lambda_T \,:\, e \in T\} \leq w(e), \text{ for all } e \in E, \\
&\lambda \geq 0, \text{ integral.}
\end{aligned}
$$

This gives a maximum integral packing of spanning trees. A combinatorial algorithm for this was given in [8], it requires $O(n^2)$ minimum cut problems. Another combinatorial algorithm was given in [42], its complexity is $O(n^3 m \log(n^2/m))$.

## 5.5 Increasing the weight of minimum spanning trees

We deal with a graph $G = (V, E)$ where each edge $e \in E$ has an original weight $w_e^0$ and we can assign to $e$ a new weight $w_e \geq w_e^0$. The *cost* of giving the weight $w_e$ is $c_e(w_e)$. The function $c_e(\cdot)$ is nondecreasing, convex, piecewise linear and $c_e(w_e^0) = 0$. We study the following problem: Given a value $\lambda \geq 0$ find a minimum cost set of weights so that the weight of a minimum spanning tree is $\lambda$.

Fredericksonand Solis-Oba [38] gave an algorithm for the case when $c_e(\cdot)$ is linear and nondecreasing, later a different derivation of their algorithm and a slight extension to deal with convex piecewise linear costs, was given in [4]. We outline this approach below.

For every edge $e$ we have a convex nondecreasing piecewise linear cost function of the weight $w_e$. This is easy to model using linear programming as follows. Assume that for every edge $e$ there are $m_e$ possible slopes $d_e^1, \dots, d_e^{m_e}$ of $c_e(\cdot)$. For the value $\bar{w}$ the cost $c_e(\bar{w})$ can be obtained as the optimal value of

$$\min \sum_k d_e^k\, x_e^k \tag{30}$$

$$\sum_k x_e^k + w_e^0 = \bar{w} \tag{31}$$

$$0 \leq x_e^k \leq u_e^k, \quad 1 \leq k \leq m_e. \tag{32}$$

We assume that $d_e^k < d_e^{k+1}$, for $k = 1, \dots, m_e - 1$. The value $u_e^k$ is the size of the interval for which the slope $d_e^k$ is valid. The solution $\bar{x}$ of this linear program is as follows:

$$\text{there is an index } k_e \geq 1 \text{ such that} \tag{33}$$

$$\bar{x}_e^k = u_e^k, \text{ for } 1 \leq k \leq k_e - 1, \tag{34}$$

$$u_e^{k_e} > \bar{x}_e^{k_e} = \bar{w} - w_e^0 - \sum_{1 \leq k \leq k_e - 1} u_e^k \geq 0, \tag{35}$$

$$\bar{x}_e^k = 0, \text{ for } k_e + 1 \leq k \leq m_e. \tag{36}$$

Thus the problem can be modeled as

$$\min dx \tag{37}$$

$$\sum_{e \in T} w_e \geq \lambda, \text{ for each tree } T \tag{38}$$

$$w_e = w_e^0 + \sum_{k=1}^{m_e} x_e^k, \text{ for each edge } e \tag{39}$$

$$0 \leq x \leq u. \tag{40}$$

A combinatorial algorithm for solving this parametric linear program was given in [4]. It uses as subroutines the strength problem, packing spanning trees and network reinforcement. The complexity of producing the primal solutions is $O(mn^5 \sum m_e)$, and the complexity of obtaining the dual solutions is $O(mn^6 \sum m_e)$.

## 5.6 Potts' model in Statistical Physics

Here we describe an application presented in [2, 77]. We start with a brief description of Potts' model from Statistical Physics, for a more complete treatment see [84]. A *lattice of spins* is a graph where each node $i$ has an associated variable $\sigma_i$. Each $\sigma_i$ can take values in $\mathbb{Z}_q = \{0, \dots, q-1\}$. Edges $ij$ are called *bonds*, and they have an associated weight $K_{ij} \geq 0$. Each configuration $\sigma = (\sigma_1, \dots, \sigma_n)$ has an energy

$$E(\sigma) = \sum_{ij} K_{ij} \delta_{\sigma_i \sigma_j}.$$

Here the sum runs over all bonds, and $\delta_{\sigma_i\sigma_j}$ takes the value $1$ if $\sigma_i = \sigma_j$, and $0$ otherwise. The partition function is

$$Z = \sum_{\sigma} \exp(E(\sigma)),$$

where the sum runs over all possible values of $\sigma \in \mathbb{Z}_q^n$. The partition function encodes the statistical properties of a system in thermodynamic equilibrium.

Using the transformation $\exp(K_{ij}) - 1 = q^{\alpha_{ij}}$, and after some algebraic manipulation the partition function can be written as

$$Z = \sum_{F} q^{c(F)+\sum_{ij\in F}\alpha_{ij}}. \tag{41}$$

Here the sum runs over all subsets $F$ of bonds, and $c(F)$ is the number of connected components of $F$, counting isolated nodes as components. As pointed out in [56], when $q$ tends to infinity the sum in (41) converges to $Nq^{f^*}$, where

$$f^* = \max_{F}\{c(F) + \sum_{ij\in F}\alpha_{ij}\}, \tag{42}$$

and $N$ is the number of optimal solutions of (42). It is easy to see that finding the maximum in (42) is equivalent to finding the minimum in (2). The value of the maximum in (42) gives the order of magnitude of the partition function, and if the weights $K_{ij}$ are arbitrary reals, then the number $N$ is likely to be one, and one can have a good approximation of the partition function value.

# 6 Survivable networks

Satisfying a suitable degree of survivability has become one of the most important issues in the design of telecommunication networks. Survivable networks must fulfill some connectivity requirement that ensure connections between parts of the network, that is networks that are still functional after the failure of certain links. Computing network topologies that provide a sufficient degree of survivability has become the main objective when designing telecommunication networks.

As fiber-optic technology provides high transmission capacity, telecommunication networks tend to be sparse, and in consequence, the failure of a single (or more) link (node) of the network might be of heavy consequences if the network does not provide alternative routing paths. This leads to the problem of designing minimum-cost telecommunication network with high reliability level, namely with sufficient routing paths between each pair of nodes.

More precisely, consider an undirected graph $G = (V, E)$ such that with each node $u \in V$ is associated a nonnegative integer $r(u)$, called its *connectivity type*, that represents its importance of communication from and to it. The edge-survivability (resp. node-survivability) conditions are then stated as the requirement of the existence of at least

$$r(s,t) = \min\{r(s), r(t)\} \tag{43}$$

edge-disjoint (resp. node-disjoint) paths in the subgraph of $G$ for any pair of nodes $s, t \in V$. Given edge costs $c(e) \in \mathbb{R}_+$, $e \in E$, the *edge (node) survivable network design problem* is to determine a minimum cost subgraph of $G$ satisfying the edge (node-) survivability conditions. This model, introduced by Grötschel and Monma [46], has received considerable attention in the past. Moreover, partition inequalities arise as valid inequalities for many variants of this model.

Expressing the survivability requirements using the connectivity types allows to model a wide variety of well-known combinatorial optimization problems which have been intensely studied for several decades. For instance, if the connectivity type vector $r = (r(u), u \in V)$ is uniform, say $r(u) = k$ for all $u \in V$ where $k$ is a positive integer, then the edge (node-) survivable network design problem is nothing but the *$k$-edge (node) connected subgraph problem*. Another variant whose underlying topology is of great interest in telecommunications is when the connectivity types are 1 and 2. Here the nodes are of two types: ordinary nodes which should be linked to the final network and important nodes with high degree of survivability. As we will see later, many classes of partition inequalities are valid for these problems. Moreover they play a central role in their resolution.

In the rest of this paper we will mostly deal with the edge version of the survivable network design problem. So we will usually omit the "edge" prefix and simply consider survivable network design problem instead (SNDP for short). Given a graph $G = (V, E)$ and $W \subseteq V$, we let $\overline{W} = V \setminus W$. For a nonempty node subset $W \subsetneq V$, the set of edges having exactly one endnode in $W$ is called a *cut* or a *cutset* and is denoted by $\delta_G(W)$. Moreover, if $s \in W$ and $t \notin W$, then $\delta(W)$ is called an *$st$-cut*. For all our notations, we don't use the subscript $G$ whenever the graph $G$ can be deduced from the context. For $F \subseteq E$, we denote by $V(F)$ the set of nodes which are spanned by the edges in $F$. For $W \subseteq V$, we denote by $E(W)$ the set of edges with both endnodes in $W$, and by $G(W) = (W, E(W))$ the subgraph induced by $W$. Given a polytope $P \subseteq \mathbb{R}^n$, the *dominant* of $P$ is the polyhedron given by $P + \mathbb{R}^n_+$.

## 6.1 Valid inequalities

In this subsection we shall present some families of valid inequalities for the SNDP. Throughout we consider a graph $G = (V, E)$ and a connectivity type vector $r \in \{0, 1, 2\}^V$.
For all $W \subseteq V, \emptyset \neq W \neq V$, $con(W) = \min\{r(W), r(V \setminus W)\}$ where $r(W) = \max\{r(u) : u \in W\}$. From Menger's theorem [68], it follows that the SNDP is equivalent

to the following integer linear program

$$\begin{aligned}
&\text{minimize} \sum_{e\in E} c(e)x(e) && \\
&\text{subject to} && \\
&x(e) \geq 0 && \text{for all } e \in E, && (44)\\
&x(e) \leq 1 && \text{for all } e \in E, && (45)\\
&x(\delta(W)) \geq con(W) && \text{for all } W \subseteq V, \emptyset \neq W \neq V, && (46)\\
&x(e) \in \{0,1\} && \text{for all } e \in E. && (47)
\end{aligned}$$

Inequalities (44) and (45) are called *trivial inequalities* and inequalities (46) are called *cut inequalities*.

It is not hard to see that the following inequalities, introduced by Grötschel et al. [48](see also [81]), are valid for the node version of the problem

$$x(\delta_{G\setminus U}(W)) \geq con_{G\setminus U}(W) - |U|, \quad \text{for all } U \subseteq V, \emptyset \neq U \neq V,\ |U| < con_{G\setminus U}(W), \quad (48)$$
$$\text{and for all } W \subseteq V \setminus U.$$

Inequalities (48) are called *node cut inequalities*. By adding these inequalities to the above integer linear program and using again Menger's theorem [68], we obtain an integer linear programming formulation for the node-SNDP.
Let us note that the cut and node cut inequalities can be separated in polynomial time using network flows. In what follows further valid inequalities induced by partitions of the underlying graph are given.

*Multicut inequalities*

Let $\{V_1, \ldots, V_p\}$ be a partition of $V$. If $con(V_i) = 1$ for $i = 1 \ldots, p$, the graph obtained from any solution to the SNDP by contracting every subgraph $G(V_i)$, $i = 1, \ldots, p$, must then be connected. Therefore, the following inequality is valid for the SNDP.

$$x(\delta(V_1, \ldots, V_p)) \geq p - 1 \qquad \text{for all partition } \{V_1, \ldots, V_p\} \text{ such that } con(V_i) = 1, \text{ for } i = 1, \ldots, p. \quad (49)$$

Inequalities of type (49) are called *multicut inequalities*. In [46], Grötschel and Monma (see also [48, 81]) showed that inequalities (49), together with the trivial inequalities (44) and (45), suffice to describe the survivable network design polytope when $r(i) = 1$ for all $i \in V$. Moreover, as mentioned before, the dominant of the spanning tree polytope is defined by (49) and the nonegativity constraints [83, 73]. This inequalities have been used inside an algorithm for the *Network Loading Problem* in [7].

For general connectivity vector $r \in \mathbb{Z}^V$ having at least one node $u \in V$ with $r(u) = 0$, Grötschel et al. [47] showed that the separation problem for inequalities (49) is NP-hard.

Furthermore, if $r(u) \geq 1$ for all $u \in V$, as mentioned by Kerivin and Mahjoub [59], inequalities (49) can then be separated in polynomial time by applying Cunningham or Barahona algorithms on the graph obtained from $G$ by contracting the set of nodes $\{u \in V : r(u) > 1\}$.

*Partition inequalities*

In [48], Grötschel et al. introduced a class of valid inequalities for SNDP($G, r$), called partition inequalities, that generalizes the cut inequalities (46). These inequalities are as follows. Let $\{V_1, \ldots, V_p\}$, $p \geq 3$, be a partition of $V$ such that $1 \leq con(V_i) \leq 2$ for $i = 1, \ldots, p$. Denote $I_2 = \{i : con(V_i) = 2,\ i = 1, \ldots, p\}$. The *partition inequality* induced by $\{V_1, \ldots, V_p\}$ is given by

$$x(\delta(V_1, \ldots, V_p)) \geq \begin{cases} p-1 & \text{if } I_2 = \emptyset, \\ p & \text{otherwise.} \end{cases} \tag{50}$$

Obviously, if all connectivity types are equal to 2, a partition inequality (50) is implied by the cut constraints $x(\delta(V_i)) \geq 2$. (We remark that considering the case where $p = 2$ gives a cut inequality (46).)

The separation problem for the partition inequalities (50) is NP-hard in general [47] (Recall that we are considering here $r \in \{0, 1, 2\}^V$). As mentioned above, Grötschel et al. [47] showed that, even in the restricted case where $r \in \{0, 1\}^V$, the separation problem remains NP-hard. If $r \in \{1, 2\}^V$, Kerivin and Mahjoub [59] proved that the separation problem associated with

$$x(\delta(V_1, \ldots, V_p)) \geq p \qquad \text{if } I_2 \neq \emptyset, \tag{51}$$

where $\{V_1, \ldots, V_p\}$ is a partition of $V$, reduces to minimizing a submodular function and therefore can be solved in polynomial time. As mentioned before, Barahona and Kerivin [11] devised a pure combinatorial algorithm, based on the submodular intersection problem, for separating inequalities (51).

*$F$-partition inequalities*

Suppose the connectivity type vector $r$ is such that $r(u) = 2$ for all $u \in V$. A class of valid inequalities for the survivable network design polytope in this case was introduced by Mahjoub [64] as follows. Consider a partition $\{V_1, \ldots, V_p\}$ of $V$ and let $F \subseteq \delta(V_1)$ with $|F|$ odd. By adding the inequalities

$$\begin{aligned} &x(\delta(V_i)) \geq 2 && \text{for } i = 2, \ldots, p, \\ &-x(e) \geq -1 && \text{for } e \in F, \\ &x(e) \geq 0 && \text{for } e \in \delta(V_1) \setminus F, \end{aligned}$$

we obtain

$$2x(\Delta) \geq 2(p-1) - |F|,$$

where $\Delta = \delta(V_1, \ldots, V_p) \setminus F$. Dividing by 2 and rounding up the right-hand side lead to

$$x(\Delta) \geq p - \left\lceil \frac{|F|}{2} \right\rceil. \tag{52}$$

Inequalities (52) are called *$F$-partition inequalities*. Note that if $|F|$ is even, the corresponding inequality (52) is then implied by inequalities (44), (45) and (46). It is straightforward that inequalities (52) remain valid for SNDP when $r \in \{0, 1, 2\}^V$ and $con(V_i) = 2$ for $i = 1, \ldots, p$.

The partition and $F$-partition inequalities are special cases of more general classes of inequalities given by Grötschel et al. [48] for SNDP (see also [81]). Furthermore, Kerivin et al. [61] considered a subclass of $F$-partition inequalities, called generalized odd-wheel inequalities, to give sufficient conditions for inequalities (52) to be facet-defining. They also introduced an extension of inequalities (52) to the case where the inducing partition $\{V_1, \ldots, V_p\}$ is such that $con(V_i) \in \{1, 2\}$ for $i = 1, \ldots, p$.

The separation problem for the $F$-partition inequalities is still an open question. However, if the sets $V_i$ of partitions are singletons, the corresponding $F$-partition inequalities are then blossom inequalities for $b$-matching which can be separated in polynomial time with the algorithm of Padberg and Rao [74]. Moreover, when the edge subset $F$ is fixed, as pointed out by Baïou et al. [5], the separation problem for inequalities (52) can be solved in polynomial time. In fact, one can delete the set of edges $F$ from $G$ and consider the resulting graph $G' = (V', E')$, say. An $F$-partition in $G$ can be written in $G'$ as

$$x(\delta_{G'}(V_1, \ldots, V_p)) \geq p - \left\lceil \frac{|F|}{2} \right\rceil, \tag{53}$$

where $V_1$ contains exactly one endnode of each edge of $F$. There are $2^{|F|}$ possibilities to assign nodes of $F$ to $V_1$. For each one we can contract the nodes of $F$ in $V_1$ and solve the separation problem for inequalities (53). As Cunningham's algorithm and Barahona's algorithm provide a most violated multicut inequality, if there is any, this can then be done in polynomial time. As it is shown in [61], $F$-partition inequalities play a central role for solving SNDP in the low connectivity case, within the framework of a cutting plane algorithm.

Consider the inequalities of type

$$x(\delta(V_1, \ldots, V_p)) \geq ap + b. \tag{54}$$

where $\{V_1, \ldots, V_p\}$ is a partition of $V$ and $a$ and $b$ are two fixed scalars. These inequalities have been the subject of a large part of the first part of the paper. And consider now the $k$-edge connected network problem, that is, the SNDP where $r(u) = k$ for all $u \in V$. Grötschel et al. [48] introduced the following inequalities

$$x(\delta(V_1, \ldots, V_p)) \geq \left\lceil \frac{kp}{2} \right\rceil, \tag{55}$$

where $\{V_1, \ldots, V_p\}$ is a partition of $V$. Inequalities (55) are clearly redundant with respect to the cut inequalities (46) if $kp$ is even. In order to have an approximate separation routine, instead of separating inequalities (55), one can separate the inequalities

$$x(\delta(V_1, \ldots, V_p)) \geq \frac{kp}{2},$$

which are nothing but inequalities (54) where $a = \frac{k}{2}$ and $b = 0$.

Let $Z \subset V$ be a node set with $|Z| = t \leq k - 1$ and $\{V_1, \ldots, V_p\}$ a partition of $V \setminus Z$. For the $k$-node connected network problem, Grötschel and Monma [46] introduced the *node partition inequalities* which are as follows

$$x(\delta_{G \setminus Z}(V_1, \ldots, V_p)) \geq \begin{cases} p - 1 & \text{if } k - t = 1, \\ \left\lceil \dfrac{p(k-t)}{2} \right\rceil & \text{if } k - t \geq 2. \end{cases} \tag{56}$$

Grötschel and Monma [46] also gave necessary and sufficient conditions for inequalities (56) to be facet-defining. If $k - t = 1$, inequalities (56) are then multicut inequalities, and therefore can be separated in polynomial time. If $k - t$ is positive and even, they are nothing but inequalities (54) and their separation is also polynomially solvable. As we mentioned for inequalities (55), one can use Baïou, Barahona and Mahjoub's algorithm for separating inequalities (54) in order to approximate the separation problem for inequalities (56) where $k - t$ is positive and odd.

## 6.2 Polyhedral consequences

We now shall discuss some polyhedral consequences of the valid inequalities introduced above. But first let us define three classes of graphs we are going to consider hereafter and in this section. A *homeomorph* of $K_4$ (i.e., the complete graph on four nodes) is a graph obtained from $K_4$ when its edges are subdivided into paths by inserting new nodes of degree two. A graph is called *series-parallel* if it contains no homeomorph of $K_4$ as a subgraph. A graph is called *outerplanar* if it can be drawn in the plane as one cycle with noncrossing chords. We note that outerplanar graphs are also series-parallel. A graph is said to be a *Halin graph* if it consists of a cycle and a tree without nodes of degree 2 whose pending nodes are precisely the nodes of the cycle.

In [64], Mahjoub showed that when $G$ is series-parallel and $r(u) = 2$ for all $u \in V$ (that is the 2-edge connected subgraph problem), the corresponding polytope is given by the trivial inequalities (44) and (45), and the cut inequalities (46). This linear description was generalized to the case where $r \in \{0, 2\}^V$ by Baïou and Mahjoub [6] as well as to the case where $r \in \{0, k\}^V$ and $k$ is even by Didi Biha and Mahjoub [27]. Kerivin and Mahjoub [60] extended those results to the more general case where the connectivity types are all even. For connectivity type vectors $r$ such that $r(u) = 2$ for all $u \in V$, Barahona and Mahjoub [12] studied the 2-edge and 2-node connected subgraph polytopes in the graphs that can be decomposed by 3-edge cutsets. (A *3-edge cutset* is a cut that consists of exactly three edges.) They showed that if a graph $G$ decomposes into $G_1$ and $G_2$ by a 3-edge cutset, the system describing the polytope is then the union of both systems describing the polytopes associated with $G_1$ and $G_2$. As a consequence, they obtained that inequalities (52) together with the trivial and cut inequalities completely describe the 2-edge connected subgraph polytope on Halin graphs for this case of connectivity type vectors. They also presented similar results for the node version. Some extensions of this work to the case where $r \in \{0, 2\}^V$ were studied in [67].

In some practical situations, one may need to use more than one link between two given nodes of a survivable network. This case can be seen as a relaxation of the survivable network problem, and is usually easier to handle. Let $P(G, r)$ be the *dominant* of the survivable network design polytope.
In [16], Chopra studied $P(G, r)$ when $r(u) = k$ for all $u \in V$ and $G$ is an outerplanar graph. For this case with $k$ odd, he showed that the following inequalities are valid for the polyhedron $P(G, r)$

$$x(\delta(V_1, \ldots, V_p)) \geq \left\lceil \frac{k}{2} \right\rceil p - 1 \qquad \text{for all partitions } \{V_1, \ldots, V_p\} \text{ of } V. \tag{57}$$

Moreover, he proved the following.

**Theorem 7** [16] *If $G = (V, E)$ is outerplanar, $r(u) = k$ for all $u \in V$ with $k$ odd, the polyhedron $P(G, r)$ is then given by the nonnegativity inequalities* (44) *and inequalities* (57).

The polyhedron $P(G, r)$ was previously studied by Cornuéjols et al. [18]. They showed that on series-parallel graphs and for $r(u) = 2$ for all $u \in V$, the polyhedron $P(G, r)$ is completely described by the nonnegativity inequalities (44) and the cut inequalities (46). In [3], Baïou showed that this result also holds if $r \in \{0, 2\}^V$. In the more general class of series-parallel graphs, Didi Biha and Mahjoub [28] (see also Didi Biha [25]) proved that inequalities (57) remain valid for the survivable network design problem where $r(u) = k$ for all $u \in V$ with $k$ odd, and called these inequalities *SP-partition inequalities* (SP stands for Series-Parellel). They also showed that inequalities (57) together with the nonnegativity inequalities (44) completely describe the polyhedron $P(G, r)$ in that case. As a consequence, they obtained that Theorem 7 also holds on series-parallel graphs, as conjectured by Chopra [16]. This conjecture was also proved independently by Chopra and Stoer [17]. We remark that inequalities (57) are a particular case of the partition inequalities (54). Therefore, a direct consequence of the result of Baïou et al. [5], inequalities (57) can be separated in polynomial time. As it is shown in [14, 24], SP-partition inequalities (57) have shown to be very utile for solving the SNDP with high survivability requirements. (We remind that those inequalities are valid for the SNDP only if the graph induced by the partition is series-parallel.) Didi Biha et al. [26] showed that, in a subclass of series-parallel graphs containing all the outerplanar graphs, the survivable network polytope is completely described by the trivial inequalities (44) and (45), the cut inequalities (46) and the partition inequalities (50) when $r \in \{1, 2\}^V$.

# 7 Critical extreme points

It is well known that the linear relaxation of a combinatorial optimization problem usually provides near optimal solution. In order to improve this solution, one has to add valid inequalities which are violated by fractional solutions. Many of these solutions may be extreme points of the linear relaxation and therefore, characterizing the extreme points, among the ones of the linear relaxation, which may be separated in polynomial time, would be of great interest for solving the whole optimization problem. This question was first studied by Fonlupt and Mahjoub [32] for the 2-edge connected network polytope. They introduced the concept of critical extreme points of the linear relaxation of the 2-edge connected subgraph polytope. In this section, we discuss these extreme points.

Consider a graph $G = (V, E)$. We denote by $P(G)$ the polytope given by the trivial inequalities (44) and (45) and the following cut inequalities

$$x(\delta(W)) \geq 2 \qquad \text{for all } W \subset V,\ W \neq \emptyset. \tag{58}$$

We observe that the polytope $P(G)$ is the linear relaxation of the 2-edge connected network polytope.

Let $\overline{x}$ be a noninteger extreme point of $P(G)$. Let $\overline{x}'$ be a solution obtained by replacing some (but at least one) noninteger components of $\overline{x}$ by $0$ or $1$ (and keeping all the other components of $\overline{x}$ unchanged). If $\overline{x}'$ is a point of $P(G)$, then $\overline{x}'$ can be written as a strict convex combination of extreme points of $P(G)$. If $\overline{y}$ is such an extreme point, then $\overline{y}$ is said to be *dominated* by $\overline{x}$, and we write $\overline{x} \succ \overline{y}$. Note that an extreme point of $P(G)$ may dominate more than one extreme point of $P(G)$. Notice also that, if $\overline{x}$ dominates $\overline{y}$, that is, $\overline{x} \succ \overline{y}$, we then have

$$\begin{aligned}
&\{e \in E \mid 0 < \overline{y}(e) < 1\} \subset \{e \in E \mid 0 < \overline{x}(e) < 1\}\\
&\{e \in E \mid \overline{x}(e) = 0\} \subseteq \{e \in E \mid \overline{y}(e) = 0\}, \text{ and}\\
&\{e \in E \mid \overline{x}(e) = 1\} \subseteq \{e \in E \mid \overline{y}(e) = 1\}.
\end{aligned}$$

The relation $\succ$ defines a partial ordering on the extreme points of $P(G)$. The minimal elements of this ordering (i.e., the extreme points $x$ for which there is no extreme point $y$ such that $x \succ y$) correspond to the integer extreme points of $P(G)$. The minimal extreme points of $P(G)$ are called extreme points of *rank* $0$. An extreme point $x$ of $P(G)$ is said to be of *rank k*, for a fixed $k$, if $x$ only dominates extreme points of rank less or equal than $k-1$ and if it dominates at least one extreme point of rank $k-1$. We notice that if $\overline{x}$ is an extreme point of $P(G)$ of rank $1$ and if we replace one fractional component of $\overline{x}$ by $1$, keeping unchanged the other components, we obtain a feasible point $\overline{x}'$ of $P(G)$ which can be written as a convex combination of integer extreme points of $P(G)$. We also observe that the extreme points of $P(G)$ may have rank at most $|V|$.

Fonlupt and Mahjoub [32] introduced the following reduction operations with respect to a solution $\overline{x}$ of $P(G)$.

$\theta_1$: Delete an edge $e$ with $\overline{x}(e) = 0$.

$\theta_2$: Contract an edge $e$ having one of its endnodes of degree 2.

$\theta_3$: Contract a node subset $W$ such that $G(W)$ is 2-edge connected and $\overline{x}(e) = 1$ for all $e \in E(W)$.

Starting from a graph $G$ and a point $\overline{x}$ of $P(G)$, let $G'$ be a reduced graph and $\overline{x}'$ be a point of $P(G')$, both obtained by applying operations $\theta_1, \theta_2, \theta_3$. It is not hard to see that $\overline{x}$ is an extreme point of $P(G)$ if and only if $\overline{x}'$ is an extreme point of $P(G')$. Moreover we have

**Lemma 8** [32] *$\overline{x}$ is an extreme point of $P(G)$ of rank 1 if and only if $\overline{x}'$ is an extreme point of $P(G')$ of rank 1.*

An extreme point of $P(G)$ is said to be *critical* [32] if it is of rank 1 and if none of the operations $\theta_1, \theta_2, \theta_3$ can be applied for it. By Lemma 8, the characterization of the extreme points of rank 1 thus reduces to those of the critical extreme points of $P(G)$. In [32], Mahjoub and Fonlupt gave the following necessary conditions for a fractional extreme point of $P(G)$ to be critical.

**Theorem 9** [32] *Let $G = (V, E)$ be a 2-edge connected graph and $\overline{x}$ a fractional extreme point of $P(G)$. If $\overline{x}$ is a critical extreme point of $P(G)$, then the following hold.*

*(i)* $V = V^1 \cup V^2$ *with* $V^1 \cap V^2 = \emptyset$,

$E = E^1 \cup E^2$ *with* $E^1 \cap E^2 = \emptyset$,

$(V^1, E^1)$ *is an odd cycle,*

$(V^1 \cup V^2, E^2)$ *is a forest whose set of pending nodes is $V^1$ and such that all the nodes in $V^1$ have degree 3,*

*(ii)* $\overline{x}(e) = \frac{1}{2}$ *for* $e \in E^1$,

$\overline{x}(e) = 1$ *for all* $e \in E^2$, *and*

*(iii)* $\overline{x}(\delta(W)) > 2$ *for all cut* $\delta(W)$ *such that* $|W| \geq 2$ *and* $|\overline{W}| \geq 2$.

**Remark 2.1** By (ii) and (iii) of Theorem 9, if $G$ supports a critical extreme point, then $G$ is 3-edge connected, and $|\delta(S)| \geq 4$ for every cut $\delta(S)$ such that $|S| \geq 2$ and $|\overline{S}| \geq 2$.

Theorem 9 has some interesting algorithmic and polyhedral consequences. We first note that operations $\theta_1, \theta_2, \theta_3$ can be performed in polynomial time and in any order. Consider now a graph $G = (V, E)$ and a critical extreme point $\bar{x}$. From Theorem 9, it follows that there exists an odd cycle $C$ of $G$ such that $\bar{x}(e) = \frac{1}{2}$ for $e \in C$ and $\bar{x}(e) = 1$ for $e \in E \setminus C$. Moreover $E \setminus C$ induces a forest whose pending nodes are precisely the nodes of $V(C)$. So the inequality

$$\sum_{e \in C} x(e) \geq \frac{|C| + 1}{2}, \tag{59}$$

which is valid for the 2-edge connected network problem, is violated by $\bar{x}$. Actually, a constraint (59) is an $F$-partition inequality (52) where $F$ is the set of leaves of the forest. Thus, by the remark above we have the following.

**Theorem 10** [32] *Critical extreme points can be separated from the 2-edge connected network polytope in polynomial time.*

Kerivin et al. [61] showed that an inequality (59) is a special case of a more general class of facet-defining inequalities for the 2-edge connected network polytope. Consequently, by Theorem 10, critical extreme points may be separated by $F$-partition facets.

The concept of critical extreme points has also been studied by Mahjoub and Nocq [66] for the 2-node connected network polytope. The following inequalities

$$x(\delta_{G\setminus v}(W)) \geq 1 \qquad \text{for all } v \in V, W \subset V \setminus \{v\}, W \neq \emptyset. \tag{60}$$

are valid for the 2-node connected network polytope. We observe that these inequalities are a special case of the node partition inequalities (56). In [66], Mahjoub and Nocq studied the polytope $Q(G)$ given by inequalities (44), (45), (58) and (60). This polytope is the linear relaxation of the 2-node connected network polytope. They extended the concept of extreme points of rank 1 and critical extreme points to the polytope $Q(G)$. They also gave necessary and sufficient conditions for an extreme point of $Q(G)$ to be critical.

We now look at the case where $r \in \{1,2\}^V$. As pointed out in [61] (see also [58]), the $F$-partition inequalities (52) can straighforwardly be extended to the case $r \in \{1,2\}^V$ as follows

$$x(\Delta) \geq p - 1 - \Big\lfloor \frac{p_1 + |F|}{2} \Big\rfloor, \tag{61}$$

where $p_1 = |\{i \mid con(V_i) = 1,\ i = 2, \ldots, p\}|$. We remark here that $|F|$ is not necessarily odd. In fact, inequalities (61) are dominated by the cut and trivial inequalities if and only if $p_1$ and $|F|$ have the same parity.

Let $R(G,r)$ be the polytope described by the trivial inequalities (44) and (45), the cut inequalities (46) and the partition inequalities (50). The interest in considering the partition inequalities (50) for $R(G,r)$ is because they can be separated in polynomial time in the case $r \in \{1,2\}^V$ as proved in [59]. Given a solution $\overline{x}$ of $R(G,r)$, the following operations, described in [61] and given with respect to $\overline{x}$, extend in a straightforward way the operation $\theta_2$, introduced above, to the case where $r \in \{1,2\}^V$.

$\theta'_1$: Contract an edge $uv$ such that $\bar{x}(uv) = 1$, $r(u) = 1$ and $\overline{x}(\delta(u)) \leq 2$.

$\theta'_2$: Contract an edge $uv$ such that $r(u) = 2$, $\delta(u) = \{uv, uw\}$ and $r(w) = 2$.

Note that these reduction operations can also be realized in polynomial time. We also notice that operation $\theta_3$, previously given for the case where $r(u) = 2$ for all $u \in V$, can be extended to the (1,2)-survivable network problem by considering node sets $W \subset V$ with $r(u) = 2$ for all $u \in W$.

With a graph obtained from $G$ by contracting an edge $e = uv \in E$, we associate the connectivity type vector $r_e \in \{1,2\}^{|V|-1}$ such that $r_e(w) = con(\{u,v\})$ and $r_e(u) = r(u)$ if $u \in V \setminus \{u,v\}$, where $w$ is the node that arises from the contraction of $e$. Let

$G' = (V', E')$ be a graph obtained by repeated applications of operations $\theta_1$, $\theta_2$, $\theta_3$, $\theta'_1$, $\theta'_2$. Denote by $r' \in \{1,2\}^{V'}$ the connectivity type vector corresponding to the graph $G'$ and by $\bar{x}'$ the restriction of $\bar{x}$ on $E'$. Kerivin et al. [61] showed that if $\bar{x}$ is an extreme point of $R(G,r)$, then $\bar{x}'$ is also an extreme point of $R(G',r')$. Moreover they proved that these operations keep trace of the validity and the violation of the inequalities of type (46), (50) or (61) in the graph $G'$ with respect to the connectivity type vector $r'$. Thus for looking for a violated inequality of type (46), (50) or (61) in $G$ with respect to $r$, one can compute a violated one in $G'$ with respect to $r'$. Moreover $G'$ can be relatively small.

Operations $\theta_1$, $\theta_2$, $\theta_3$, $\theta'_1, \theta'_2$) have been used by Kerivin et al. [61] in a preprocessing phase of a cutting plane algorithm for the (1,2)-survivable network design problem (respectively the 2-node connected network problem), and have shown to be very effective for solving these problems.

# 8 Survivability with length constraints

In general, survivability requirement is not sufficient to guarantee a cost effective routing. Indeed, the alternative routing paths may be too long and then, too costly to be suitable. In consequence, further technical constraints have to be added, in particular one can impose a limit on the length of the rerouting paths. In order to limit the rerouting, one thus must have at least two edges (node)-disjoint paths with bounded length between each pair origin-destination, so that if one of the paths fails, the traffic may be rerouted (in a minimum time) on the second one. In many practical situations, the length of the routing path is considered as the number of links (also called hops) in the path, and then we talk about *hop-constrained path*. In this section, we shall discuss valid partition inequalities for some variants of the constrained survivable network design problems.

## 8.1 Survivability with bounded rings

In [35], Fortz et al. considered the problem of designing a minimum cost 2-node connected network such that each edge belongs to a cycle of a bounded length. This problem can be presented as follows: Given a graph $G = (V,E)$ such that each edge $e \in E$ has a cost $c(e)$ and a length $d(e)$, and a positive integer $L$, the problem consists of finding a minimum cost 2-node connected subgraph $(W,F)$ such that each edge of $F$ belongs to a cycle of length less or equal than $L$. In [34], Fortz and Labbé provided classes of valid inequalities and discussed the associated separation problems. In particular they introduced the inequalities

$$x(\delta(V_1,\dots,V_p)) \geq 2M(p,L) \tag{62}$$

where $M(p,L) = n+ \min\{\lceil \frac{p-L}{L-2} \rceil, \lceil \frac{p}{L-1} \rceil\}$. They approximate $M(p,L)$ with $a(p-1)$ where $a = \frac{nL}{(n-1)(L-1)}$. They showed empirical evidence that this is a good approximation. They also reported computational results obtained with a cutting plane algorithm. For a complete survey of this problem, see Fortz [33].

In [36], Fortz et al. studied the edge version of the above problem, the 2*-edge connected subgraph with bounded rings problem (2ECSBR)*. They considered the case where the length of each edge is 1. So the problem here is to find a minimum cost 2-edge connected subgraph such that each edge belongs to a cycle with no more than $L$ edges. Fortz et al. [36] introduced a class of valid inequalities, and, using this, they gave an integer programming formulation for the problem in the space of the design variables. In what follows we describe these inequalities.

Let $G = (V, E)$ be a graph and $L \geq 3$. If $\pi = \{V_0, \ldots, V_p\}$ is a partition of $V$, then we let $C_\pi = \cup_{i=0}^{p-1}[V_i, V_{i+1}] \cup [V_0, V_p]$ and $T_\pi = \delta(V_0, \ldots, V_p) \setminus C_\pi$. Suppose now that the partition $\pi$ is such that $p \geq L$ and let $e \in [V_0, V_p]$. Consider the inequality

$$x(T_\pi^e) \geq x_e \tag{63}$$

where

$$T_\pi^e = T_\pi \cup ([V_0, V_p] \setminus \{e\}). \tag{64}$$

Fortz et al. [36] showed that inequalities (63) are valid for the polytope associated with the 2ECSBR. Inequalities (63) are called *cycle inequalities*. Moreover they proved that trivial, cut and cycle inequalities together with the integrality constraints yiels an integer programming formulation for the 2ECSBR. Moreover, by adding the constraints

$$x(\delta_{G-v}(W)) \geq 1, \quad W \subset V \setminus \{v\},\, v \in V,$$

we obtain a formulation for the 2-node connected subgraph with bounded rings problem (when the lengths are equal to 1).

It is not hard to see that the separation problem for inequalities (63) associated with an edge $e = st$ reduces to finding a minimum weight edge subset that intersects all $st$-paths of length $\leq L - 1$. Fortz et al. [36] (see also [65]) showed that, when $L \leq 4$, this problem reduces to a max-flow problem in an appropriate directed graph and hence can be solved in polynomial time. As a consequence, they obtained a polynomial time separation algorithm for inequalities (63) when $L \leq 4$. In what follows we present their algorithm.

First it can be easily shown ( see Fortz et al. [36]) that the separation problem for inequalities (63) can be solved in polynomial time for a $0 - 1$ solution $\bar{x}$.
Now suppose that the solution contains fractional values. Let $G = (V, E)$ be a graph and $s, t$ two nodes of $V$. Given a positive integer $B$, define an $(s, t)$-$B$*-path cut* to be any edge set $C$ of $E$ that intersects every $st$-path of $G$ with at most $B$ edges. Given a weight vector $w \in \mathbb{R}_+^m$, the minimum $(s, t)$-$B$-path cut problem (BPCP) is to find an $(s, t)$-$B$-path cut of minimum weight. Fortz et al. [36] showed that the separation problem for inequalities (63) reduces to solving BPCP for every edge $e = st \in E$ and $B = L - 1$ with respect to the weight vector $\bar{x}$.

Observe that if $B = 2$, finding a minimum $(s, t)$-2-path cut reduces to finding a minimum cut separating $s$ and $t$ in the graph induced by $s$, $t$ and the nodes adjacent to both $s$ and $t$.

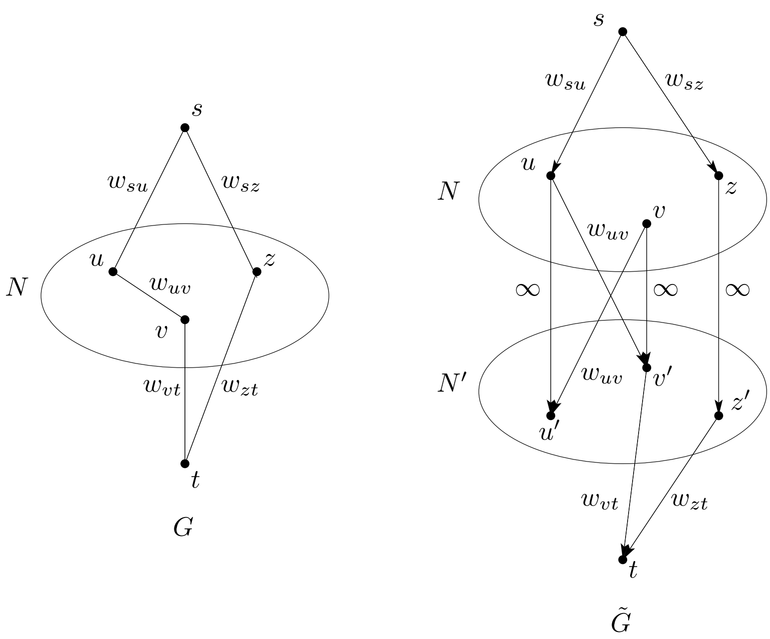


Figure 1: Construction of $\tilde{G}$

For $B = 3$, they showed that the minimum $(s,t)$-$B$-path cut problem reduces to a maximum flow problem, and can then be solved in polynomial time.
First, note that any node $u$ of $V$ which is not adjacent neither to $s$ nor to $t$ cannot belong to an $st$-path of length at most 3 and so can be deleted. So we may assume that $G$ does not contain such nodes.

Now the idea consists in constructing a directed graph $\tilde{G} = (\tilde{N}, \tilde{A})$ from the original one as follows. Let $N = V \setminus \{s,t\}$. Let $N'$ be a disjoint copy of $N$ (where we denote the copy of $u \in N$ that is in $N'$ by $u'$), and set $\tilde{N} = \{s\} \cup \{t\} \cup N \cup N'$. For each edge $su \in E$ with weight $w_{su}$ make arc $(s,u) \in \tilde{A}$ with capacity $w_{su}$, for each edge $vt \in E$ make arc $(v',t) \in \tilde{A}$ with capacity $w_{vt}$, and for each edge $uv \in E$ with $u, v \notin \{s,t\}$, make arcs $(u,v')$ and $(v,u')$, both with capacity $w_{uv}$. For each $u \in N$ with $u \neq s,t$ make an arc $(u,u') \in \tilde{A}$ with an infinite capacity (see Figure 1 for an illustration). Note that there is a 1-1 correspondence between the $st$-paths of length $\leq 3$ in $G$ and the $st$-directed paths of length 3 in $\tilde{G}$.

It has been shown in [36] (see also [65]) that to each $(s,t)$-3-path cut in G corresponds a finite capacity cut in $\tilde{G}$ separating $s$ and $t$. Moreover, both cuts have the same weight. Which implies that one can solve the BPCP for $B = 3$ in $G$ by solving a maximum flow problem in $\tilde{G}$.

In [36], Fortz et al. also extended inequalities ( 62) to 2ECSBR. They proved that the inequalities

$$x(\delta(V_0,\ldots,V_p)) \geq \lceil \frac{Lp}{L-1} \rceil, \tag{65}$$

are valid for 2ECSBR. To separate inequalities (65), Fortz et al. [36] developed a heuristic based on Barahona's algorithm [7] for separating the partition inequalities

$$x(\delta(V_0,\,\ldots,\,V_p)) \geq p. \tag{66}$$

Consider the following inequalities obtained from inequalities (65) by deleting the upper integral part from the right hand side.

$$x(\delta(V_0,\,\ldots,\,V_p)) \geq \frac{Lp}{L-1}. \tag{67}$$

Clearly, inequalities (67) are of type (66) (it suffices to set $x' = \frac{L-1}{L}x$). Moreover, if (67) is violated, then (65) is so. However, it may that all the inequalities of type (67) are satisfied whereas some inequalities of type (65) are violated. In order to strengthen inequalities (67), Fortz et al. [36] considered the inequalities

$$x(\delta(V_0,\,\ldots,\,V_p)) \geq \frac{Lp}{L-1} + \epsilon \tag{68}$$

with $\epsilon = \frac{p}{100n}$ and used the same algorithm of Barahona to separate these inequalities. (Here inequalities (68) can be transformed to inequalities of type (66) by setting $x' = (\frac{100nL+L-1}{100n(L-1)})x$.) As shown in [36], these algorithmic transformations have been very effective for solving the 2ECSBR.

## 8.2 Hop-constrained paths

The closely related and basic routing *hop-constrained path problem* has also seen a particular attention last years. This problem consists of finding between two distinguished nodes $s$ and $t$ a minimum cost path with no more than $L$ edges when $L$ is fixed. The *$L$-path polytope*, denoted by $L$PP($G$) is the convex hull of the incidence vectors of the $st$-paths having no more than $L$ edges. Clearly, the following inequalities are valid for $L$PP($G$).

$$x(\delta(W)) \geq 1, \quad \text{for all } st\text{-cut } \delta(W), \tag{69}$$

and are called *st-cut inequalities*. In [20], Dahl considered the dominant of the $L$-path polytope, that is the polyhedron $L$PP($G$)+$\mathbb{R}^E_+$. He described a class of valid inequalities for the problem and gave a complete description of that polyhedron when $L \leq 3$. In particular, he introduced a class of valid inequalities as follows.

Let $\{V_0,\,V_1,\,\ldots,\,V_{L+1}\}$ be a partition of $V$ such that $s \in V_0$, $t \in V_{L+1}$ and $V_i \neq \emptyset$ for all $i = 1,\ldots,L$. Let $T$ be the set of edges $e = uv$ where $u \in V_i$, $v \in V_j$ and $|i-j| > 1$. Then the inequality

$$x(T) \geq 1, \tag{70}$$

is valid for the $L$-path polyhedron. Using the same partition, this inequality can be generalized in a straightforward way as follows to the case when $k$ edge-disjoint paths are required between $s$ and $t$

$$x(T) \geq k. \tag{71}$$

Inequalities (70) and (71) are called *L-path cut inequalities* (or *jump inequalities* [21]). The separation problem for these inequalities can be solved in polynomial time, if $L \leq 3$. In fact, it is easily seen that this problem reduces to finding a minimum edge set that intersects all the $st$-paths with no more than $L$ edges. Since $L \leq 3$, as it has been shown by Fortz et al. [36], this can be done in polynomial time. Dahl [20] showed that inequalities (70) together with inequalities (69) and the nonnegativity inequalities completely describe the $L$-path polyhedron when $L \leq 3$.

In [21], Dahl and Gouveia considered the directed hop-constrained path problem. Note that the $st$-cut inequalities (69) and the $L$-path-cut inequalities (70), (71) can be easily extended to that problem. Dahl and Gouveia [21] described a class of valid inequalities obtained by lifting from the directed $L$-path-cut inequalities and showed that these inequalities together with the flow conservation constraints and the trivial inequalities characterize the directed $L$-path polytope when $L \leq 3$. They also identified valid inequalities and addressed some polyhedral issues for the case when $L \geq 4$.

A more general network design problem with hop-constraints, that has also been investigated is *the hop-constrained network design problem (HCNDP)*. This can be presented as follows: Given a graph $G = (V, E)$ with weights on the links, a set of pairs of terminals and two positive integers $k$ and $L$, find a minimum weight subgraph such that between each pair of terminals there are at least $k$ edge-disjoint paths with no more than $L$ links. This problem is NP-hard even when $k = 1$ and $L = 2$ [23].

In [53], Huygens et al. studied the HCNDP in the case when there is only one pair of terminals, say $s$ and $t$, $k = 2$ and $L = 3$. They gave an integer programming formulation for the problem in this case in the space of the design variables. They showed that the $st$-cut inequalities (inequalities (69) with right hand side 2) and the $L$-path inequalities (71) (with $k = 2$) together with the $0 - 1$ integrality constraints formulate this problem, and they gave an extension of this formulation to the case where $k \geq 2$. They also discussed the polytope $P(G, L)$ given by the constraints of the linear relaxation of this formulation. In particular, they proved that $P(G, L)$ is integral, if $L \leq 3$. This Theorem implies that the associated polytope is equal to $P(G, L)$. This result has been generalized by Dahl et al. [22] for $L = 2$ and $k$ arbitrary. Recently Bendali et al. [13] extended this to the case where $L = 3$ and $k$ arbitrary.

In addition, since the separation problem for the $st$-cut and $L$-path cut inequalities can be solved in polynomial time when $L \leq 3$, it follows that the HCNDP when $L \leq 3,\ k = 2$ and only one pair of terminals is considered can be solved in polynomial time using a cutting plane algorithm. As pointed out in [53], the formulation given above (for the HCNDP when $L \leq 3$, $k = 2$ and only one pair of terminal is considered) is no longer valid for the problem if $L \geq 4$. However for $L \leq 3$, one can see that this formulation can be easily extended to the HCNDP with arbitrary number of pairs of terminals. Huygens et al. [52] studied this variant with multiple pairs of terminals of the HCNDP when $k = 2$ and $L = 3$. They gave several classes of partitions inequalities valid for the associated polytope. They also derived separation routines, and developed a Branch-and-Cut algorithm based on these inequalities. Diarrassouba [24] studied the more general case when $L = 2$ and $k$ arbitrary. He gave different formulations of the problem, based on a transformation of the initial undirected graph into directed graphs. He also investigated the assciated polytope and identified classes of valid inequalities. Using this he devised cutting planes based algorithms for the problem.

# References


[1] R. K. Ahuja, J. B. Orlin, C. Stein, and R. E. Tarjan. Improved algorithms for bipartite network flow. *SIAM J. Comput.*, 23(5):906–933, 1994.

[2] J.-C. Anglès d'Auriac, F. Iglói, and A. Sebő. Optimal cooperation and submodularity for computing Potts' partition functions with a large number of states. *J. Phys. A*, 35(33):6973–6983, 2002.

[3] M. Baïou. On the dominant of the steiner 2-edge connected subgraph polytope. *Discrete Appl. Math.*, 112:3–10, 2001.

[4] M. Baïou and F. Barahona. A linear programming approach to increasing the weight of all minimum spanning trees. *Networks*, 52(4):227–234, 2008.

[5] M. Baïou, F. Barahona, and A. R. Mahjoub. Separation of partition inequalities. *Math. Oper. Res.*, 25(2):243–254, 2000.

[6] M. Baïou and A. R. Mahjoub. Steiner 2-edge connected subgraph polytopes on series-parallel graphs. *SIAM J. Discret. Math.*, 10:505–514, July 1997.

[7] F. Barahona. Separating from the dominant of the spanning tree polytope. *Op. Research Letters*, 12:201–203, 1992.

[8] F. Barahona. Packing spanning trees. *Math. Oper. Res.*, 20(1):104–115, 1995.

[9] F. Barahona. On the $k$-cut problem. *Oper. Res. Lett.*, 26(3):99–105, 2000.

[10] F. Barahona. Network reinforcement. *Math. Prog.*, 105:181–200, 2006.

[11] F. Barahona and H. Kerivin. Separation of partition inequalities with terminals. *Discrete Optim.*, 1(2):129–140, 2004.

[12] F. Barahona and A. R. Mahjoub. On two-connected subgraph polytopes. *Discrete Math.*, 147:19–34, December 1995.

[13] F. Bendali, I. Diarrassouba, A. Mahjoub, and J. Mailfert. The edge-disjoint 3-hop-constrained paths polytope. *Discrete Optimization*, 7(4):222 – 233, 2010.

[14] F. Bendali, I. Diarrassouba, A. R. Mahjoub, M. Didi Biha, and J. Mailfert. A branch-and-cut algorithm for the k-edge connected subgraph problem. *Networks*, 55:13–32, January 2010.

[15] E. Cheng and W. H. Cunningham. A faster algorithm for computing the strength of a network. *Inf. Process. Lett.*, 49:209–212, 1994.

[16] S. Chopra. The k-edge-connected spanning subgraph polyhedron. *SIAM J. Discret. Math.*, 7:245–259, May 1994.

[17] S. Chopra and M. Stoer. Private communication.

[18] G. Cornuéjols, J. Fonlupt, and D. Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Math. Prog.*, 33:1–27, 1985. 10.1007/BF01582008.

[19] W. H. Cunningham. Optimal attack and reinforcement of a network. *J. of ACM*, 32:549–561, 1985.

[20] G. Dahl. Notes on polyhedra associated with hop-constrained paths. *Oper. Res. Lett.*, 25(2):97 – 100, 1999.

[21] G. Dahl and L. Gouveia. On the directed hop-constrained shortest path problem. *Oper. Res. Lett.*, 32(1):15 – 22, 2004.

[22] G. Dahl, D. Huygens, A. R. Mahjoub, and P. Pesneau. On the edge-disjoint 2-hop-constrained paths polytope. *Oper. Res. Lett.s*, 34(5):577 – 582, 2006.

[23] G. Dahl and B. Johannessen. The 2-path network design problem. Report 292, Universtetet i Oslo, Institutt for informatikk, 2000.

[24] I. Diarrassouba. *Survivable Network Design Problems with High Connectivity Requirements*. PhD dissertation, Université Blaise Pascal, Clermont-Ferrand, France, 2009.

[25] M. Didi Biha. *Graphes k-arête connexes et polyèdres*. PhD dissertation, Université de Rennes 1, Rennes, France, 1998.

[26] M. Didi Biha, H. Kerivin, and A. R. Mahjoub. On the polytope of the (1,2)-survivable network design problem. *SIAM J. Discret. Math.*, 22:1640–1666, 2008.

[27] M. Didi Biha and A. Mahjoub. Steiner k-edge connected subgraph polyhedra. *Journal of Combinatorial Optimization*, 4:131–144, 2000.

[28] M. Didi Biha and A. R. Mahjoub. k-edge connected polyhedra on series-parallel graphs. *Oper. Res. Lett.*, 19(2):71 – 78, 1996.

[29] W. Dinkelbach. On nonlinear fractional programming. *Management Sci.*, 13:492–498, 1967.

[30] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In R. K. Guy, E. Milner, and N. Sauer, editors, *Combinatorial Structures and their Applications*, pages 69–87, New York, 1970. Gordon and Breach.

[31] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. In *Studies in integer programming (Proc. Workshop, Bonn, 1975)*, pages 185–204. Ann. of Discrete Math., Vol. 1. North-Holland, Amsterdam, 1977.

[32] J. Fonlupt and A. R. Mahjoub. Critical extreme points of the 2-edge connected spanning subgraph polytope. *Math. Prog.*, 105:289–310, 2006.

[33] B. Fortz. *Design of Survivable Networks with Bounded Rings*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[34] B. Fortz and M. Labbé. Polyhedral results for two-connected networks with bounded rings. *Math. Prog.*, 93:27–54, 2002.

[35] B. Fortz, M. Labbé, and F. Maffioli. Solving the two-connected network with bounded meshes problem. *Oper. Res.*, 48:866–877, 2000.

[36] B. Fortz, A. R. Mahjoub, S. T. McCormick, and P. Pesneau. Two-edge connected subgraphs with bounded rings: Polyhedral results and branch-and-cut. *Math. Prog.*, 105:85–111, 2006.

[37] A. Frank. On the edge-connectivity algorithm of Nagamochi and Ibaraki. Technical Report, Laboratoire Artemis. IMAG. Universite J. Fourier, Grenoble. France, 1994.

[38] G. N. Frederickson and R. Solis-Oba. Increasing the weight of minimum spanning trees. *J. Algorithms*, 33(2):244–266, 1999.

[39] S. Fujishige. *Submodular functions and optimization*, volume 58 of *Annals of Discrete Mathematics*. Elsevier B. V., Amsterdam, second edition, 2005.

[40] S. Fujishige and X. Zhang. New algorithms for the intersection problem of submodular systems. *Japan J. Indust. Appl. Math.*, 9(3):369–382, 1992.

[41] H. N. Gabow. Algorithms for graphic polymatroids and parametric $\overline{s}$-sets. *J. Algorithms*, 26(1):48–86, 1998.

[42] H. N. Gabow and K. S. Manu. Packing algorithms for arborescences (and spanning trees) in capacitated graphs. *Math. Prog. Ser. B*, 82:83–109, 1998.

[43] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989.

[44] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.*, 35(4):921–940, 1988.

[45] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993.

[46] M. Grötschel and C. L. Monma. Integer polyhedra arising from certain network design problems with connectivity constraints. *SIAM J. Discret. Math.*, 3:502–523, 1990.

[47] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Oper. Res.*, 40:309–330, 1992.

[48] 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):474–504, 1992.

[49] D. Gusfield. Connectivity and edge-disjoint spanning trees. *Inform. Process. Lett.*, 16(2):87–89, 1983.

[50] D. Gusfield. Computing the strength of a graph. *SIAM J. Comput.*, 20(4):639–654, 1991.

[51] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a directed graph. *J. Algorithms*, 17(3):424–446, 1994. Third Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, FL, 1992).

[52] D. Huygens, M. Labbé, A. R. Mahjoub, and P. Pesneau. The two-edge connected hop-constrained network design problem: Valid inequalities and branch-and-cut. *Networks*, 49:116–133, 2007.

[53] D. Huygens, A. R. Mahjoub, and P. Pesneau. Two edge-disjoint hop-constrained paths and polyhedra. *SIAM J. Discret. Math.*, 18:287–312, 2005.

[54] H. Imai. Network-flow algorithms for lower-truncated transversal polymatroids. *J. Oper. Res. Soc. Japan*, 26(3):186–211, 1983.

[55] M. Iri and S. Fujishige. Use of matroid theory in operations research, circuits and systems theory. *Internat. J. Systems Sci.*, 12(1):27–54, 1981.

[56] R. Juhász, H. Rieger, and F. Iglói. Random-bond potts model in the large-q limit. *Phys. Rev. E*, 64(5):056122, Oct 2001.

[57] M. Jünger and W. R. Pulleyblank. New primal and dual matching heuristics. *Algorithmica*, 13:357–380, 1995.

[58] H. Kerivin. *Réseaux fiables et polyèdres*. PhD dissertation, Université Blaise Pascal, Clermont-Ferrand, France, 2000.

[59] H. Kerivin and A. R. Mahjoub. Separation of partition inequalities for the $(1, 2)$-survivable network design problem. *Oper. Res. Lett.*, 30(4):265–268, 2002.

[60] H. Kerivin and A. R. Mahjoub. On survivable network polyhedra. *Discrete Mathematics*, 290(2-3):183 – 210, 2005.

[61] H. Kerivin, A. R. Mahjoub, and C. Nocq. $(1, 2)$-survivable networks: facets and branch-and-cut. *in The Sharpest Cut : The Impact of Manfred Padberg and His Work, MPS/SIAM Ser. Optim., M. Grötschel, ed., SIAM, Philadelphia*, pages 121–152, 2004.

[62] V. Kolmogorov. A faster algorithm for computing the principal sequence of partitions of a graph. *Algorithmica*, 56(4):394–412, 2010.

[63] L. Lovász. Submodular functions and convexity. In *Mathematical programming: the state of the art (Bonn, 1982)*, pages 235–257. Springer, Berlin, 1983.

[64] A. R. Mahjoub. Two-edge connected spanning subgraphs and polyhedra. *Math. Prog.*, 64:199–208, 1994.

[65] A. R. Mahjoub and S. T. McCormick. Max flow and min cut with bounded-length paths: complexity, algorithms, and approximation. *Math. Program.*, 124:271–284, July 2010.

[66] A. R. Mahjoub and C. Nocq. On the linear relaxation of the 2-node connected subgraph polytope. *Discrete Applied Mathematics*, 95(1-3):389 – 416, 1999.

[67] A. R. Mahjoub and P. Pesneau. On the steiner 2-edge connected subgraph polytope. *RAIRO - Oper. Res.*, 42:259–283, 2008.

[68] K. Menger. Zur allgemeinen kurventheorie. *Fundamanta Mathematicae*, 10:96–115, 1927.

[69] H. Nagamochi and T. Ibaraki. Computing edge connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Mathematics*, 5:54–66, 1992.

[70] H. Narayanan. The principal lattice of partitions of a submodular function. *Linear Algebra Appl.*, 144:179–216, 1991.

[71] H. Narayanan. *Submodular functions and electrical networks*, volume 54 of *Annals of Discrete Mathematics*. North-Holland Publishing Co., Amsterdam, 1997.

[72] H. Narayanan, S. Roy, and S. Patkar. Approximation algorithms for Min-$k$-overlap problems using the principal lattice of partitions approach. *J. Algorithms*, 21(2):306–330, 1996.

[73] C. S. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. London Math. Soc.*, 36:445–450, 1961.

[74] M. W. Padberg and M. R. Rao. Odd minimum cutsets and b-matching. *Math. Oper. Res.*, 7:67–80, 1982.

[75] S. Patkar and H. Narayanan. Principal lattice of partitions of submodular functions on graphs: Fast algorithms for principal partition and generic rigidity. In T. Ibaraki, Y. Inagaki, K. Iwama, T. Nishizeki, and M. Yamashita, editors, *Algorithms and Computation*, volume 650 of *Lecture Notes in Computer Science*, pages 41–50. Springer Berlin / Heidelberg, 1992.

[76] S. B. Patkar and H. Narayanan. Fast on-line/off-line algorithms for optimal reinforcement of a network and its connections with principal partition. *J. Comb. Optim.*, 7(1):45–68, 2003.

[77] M. Preissmann and A. Sebő. Graphic submodular function minimization: a graphic approach and applications. In *Research trends in combinatorial optimization*, pages 365–385. Springer, Berlin, 2009.

[78] M. Queyranne. A combinatorial algorithm for minimizing symmetric submodular functions. In *Proc. 6th ACM-SIAM Symp. Discrete Algorithms*, pages 98–101, 1995.

[79] R. Ravi and A. Sinha. Approximating $k$-cuts using network strength as a Lagrangean relaxation. *European J. Oper. Res.*, 186(1):77–90, 2008.

[80] J. Roskind and R. E. Tarjan. A note on finding minimum-cost edge-disjoint spanning trees. *Math. Oper. Res.*, 10(4):701–708, 1985.

[81] M. Stoer. *Design of survivable networks*. Number 1531 in Lecture Notes in Mathematics. Springer-Verlag, Berlin, 1992.

[82] M. Stoer and F. Wagner. A simple min-cut algorithm. *J. ACM*, 44(4):585–591, 1997.

[83] W. T. Tutte. On the problem of decomposing a graph into $n$ connected factors. *J. London Math. Soc.*, 36:221–230, 1961.

[84] F. Y. Wu. The Potts model. *Rev. Modern Phys.*, 54(1):235–268, 1982.