

B. Fortz · M. Labbé

Polyhedral results for two-connected networks with bounded rings

Abstract. We study the polyhedron associated with a network design problem which consists in determining at minimum cost a two-connected network such that the shortest cycle to which each edge belongs (a “ring”) does not exceed a given length K .

We present here a new formulation of the problem and derive facet results for different classes of valid inequalities. We study the separation problems associated to these inequalities and their integration in a Branch-and-Cut algorithm, and provide extensive computational results.

1. Introduction

We consider the problem of designing a minimum *cost* network N with the following constraints:

1. The network N contains at least two node-disjoint paths between every pair of nodes (*2-connectivity constraints*),
and
2. each edge of N belongs to at least one cycle whose *length* is bounded by a given constant K (*ring constraints*).

This problem is called the *Two-Connected Network with Bounded Rings (2CNBR) problem*. It was first studied by Fortz et al. [6].

In recent years, many works on telecommunication network design have focused on *low-connectivity constrained* network design problems, in which one wants to design a network with a minimum prespecified number of (edge or node) disjoint paths between any pair of nodes. The polyhedra associated with these problems have been widely studied (see e.g. Grötschel and Monma [9], Grötschel et al. [12], Stoer [15], Mahjoub [13], Biha and Mahjoub [4], and others. An extensive survey is available in Fortz [5]).

Institut d'Administration et de Gestion, Université Catholique de Louvain, Place des Doyens, B-1348 Louvain-la-Neuve, Belgium, e-mail: fortz@poms.ucl.ac.be

Institut de Statistique et de Recherche Opérationnelle, Service de Mathématiques de la Gestion, CP 210/01, Univ. Libre de Bruxelles. Boulevard du Triomphe, B-1050 Bruxelles, Belgium, e-mail: mllabbe@smg.ulb.ac.be

As pointed out in Fortz et al. [6], the ring constraints are a relevant extension as the optimal two-connected network turns out to be often a Hamiltonian cycle. Although they present valid inequalities for this problem, little is known about the structure of the associated polyhedron. Fortz et al. propose an extended formulation of the problem that is projected on the space of design variables. We introduce here an alternate formulation based only on design variables, and then study several classes of valid inequalities for which we present facet results. The formulation is presented in Section 2, where we also study the dimension of the polytope and trivial facets. In Section 3, we study cut constraints that impose two-edge connectivity. These constraints are strengthened when they are not facet inducing, leading to ring-cut inequalities that are presented in Section 4. In Section 5, we study when subset inequalities (coming from the formulation of Section 2) are facet-defining and their relation with node-cut inequalities. The last inequalities studied are node-partition inequalities in Section 6. The separation problems coming from all these inequalities are studied in Section 7 and are embedded in a branch-and-cut algorithm. Feasibility checking is also an important issue discussed in this section. Results of our experiments with the branch-and-cut algorithm are presented in Section 8.

2. Mathematical formulation and associated polyhedron

We present here a new mathematical formulation of the 2CNBR problem based on a set covering approach. We first introduce some notation.

Given a graph $G = (V, E)$ and $W \subset V$, the edge set

$$\delta(W) := \{\{i, j\} \in E \mid i \in W, j \in V \setminus W\}$$

is called the *cut* induced by W . We write $\delta_G(W)$ to make clear — in case of possible ambiguities — with respect to which graph the cut induced by W is considered. The *degree* of a node v is the cardinality of $\delta(v)$. The set

$$E(W) := \{\{i, j\} \in E \mid i \in W, j \in W\}$$

is the set of edges having both end nodes in W . We denote by $G(W) = (W, E(W))$ the subgraph induced by the edges having both end nodes in W . If $E(W)$ is empty, W is an *independent set*. G/W is the graph obtained from G by contracting the nodes in W to a new node w (retaining parallel edges).

We denote by $V - z := V \setminus \{z\}$ and $E - e := E \setminus \{e\}$ the subsets obtained by removing one node or one edge from the set of nodes or edges. $G - z$ denotes the graph $(V - z, E \setminus \delta(\{z\}))$, i.e. the graph obtained by removing a node z and its incident edges from G . This is extended to a subset $Z \subset V$ of nodes by the notation $G - Z := (V \setminus Z, E \setminus (\delta(Z) \cup E(Z)))$.

Each edge $e := \{i, j\} \in E$, has a *fixed cost* $c_e := c_{ij} \geq 0$ representing the cost of establishing the direct link connection, and a length $d_e \geq 0$. The cost of

a network $N = (V, F)$, where $F \subseteq E$ is a subset of possible edges, is denoted by $c(F) := \sum_{e \in F} c_e$. The *distance* between two nodes i and j in this network is denoted by $d_F(i, j)$ and is given by the length of the shortest path between these two nodes in F . In E , the distance is simply denoted $d(i, j) := d_E(i, j)$.

A useful tool to analyze feasible solutions of 2CNBR is the *restriction of a graph to bounded rings*. Given a graph $G = (V, E)$ and a constant $K > 0$, we define for each subset of edges $F \subseteq E$ its restriction to bounded rings F_K as

$$F_K := \left\{ e \in F : \begin{array}{l} e \text{ belongs to at least one cycle} \\ \text{of length less than or equal to } K \text{ in } F \end{array} \right\}.$$

The subgraph $G_K = (V, E_K)$ is the *restriction of G to bounded rings*. Note that an edge $e \in E \setminus E_K$ will never belong to a feasible solution of 2CNBR.

In order to formulate the 2CNBR problem, we associate with every subset $F \subseteq E$ an *incidence vector* $x^F = (x_e^F)_{e \in E} \in \{0, 1\}^{|E|}$ by setting

$$x_e^F := \begin{cases} 1 & \text{if } e \in F, \\ 0 & \text{otherwise.} \end{cases}$$

Conversely, each vector $x \in \{0, 1\}^{|E|}$ induces a subset

$$F^x := \{e \in E \mid x_e = 1\}.$$

Further we denote by $\mathcal{D}_{G,K}$ the set of incidence vectors x^F with $F \subseteq E$ such that

1. F is two-connected,
2. $F = F_K$.

Then, the 2CNBR problem consists in

$$\min \left\{ \sum_{e \in E} c_e x_e : x \in \mathcal{D}_{G,K} \right\}.$$

As we show in Section 7.1, checking that G_K is two-connected, i.e. that $\mathcal{D}_{G,K}$ is nonempty, can be done in polynomial time. We therefore assume in the remainder of this paper that there always exists a feasible solution to the problem.

Since all costs c_e , $e \in E$ are assumed to be nonnegative, there always exists an optimal solution of 2CNBR whose induced graph is minimal with respect to inclusion. More precisely, if F_K is two-connected, as $F \supseteq F_K$, F is also two-connected and the cost of F is greater than or equal to the cost of F_K . We can thus relax the constraints and just require that F_K is two-connected for a set of edges F to be feasible. Hence, 2CNBR can be equivalently formulated as

$$\min \left\{ \sum_{e \in E} c_e x_e : x \in \{0, 1\}^{|E|} \text{ and } F_K^x \text{ is two-connected} \right\}.$$

We denote by

$$\mathcal{P}_{G,K} := \text{conv}\{x \in \{0,1\}^{|E|} : F_K^x \text{ is two-connected}\}$$

the polyhedron associated to the 2CNBR problem.

For any subset of edges $F \subseteq E$ we define

$$x(F) := \sum_{e \in F} x_e.$$

We also denote by e_i the i -th unit vector in \mathbb{R}^n .

If a subset of edges $S \subseteq E$ is such that $(G - S)_K$ is not two-connected, then $G - S$ does not contain a feasible solution, and therefore each feasible solution contains at least one edge from S . As we are only interested in minimal feasible solutions, this is sufficient to formulate the 2CNBR problem as the following integer linear program :

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & x(S) \geq 1 \quad S \subseteq E, (G - S)_K \text{ is not two-connected}, \quad (1) \\ & x_e \in \{0,1\} \quad e \in E. \quad (2) \end{aligned}$$

Constraints (1) are called *subset constraints*.

Proposition 1. *Constraints (1) and (2) provide a valid formulation of the 2CNBR problem.*

Proof. Let F be a feasible solution to the 2CNBR problem and x be its incidence vector. If $(G - S)_K$ is not two-connected $G - S$ does not contain a feasible solution. Therefore, F contains at least one edge belonging to S . We can conclude that (1) are valid inequalities for the 2CNBR problem.

Conversely, suppose that x is an optimal solution to the integer linear program, and that it is not a feasible solution of the 2CNBR problem, i.e. F_K^x is not two-connected. Then the subset inequality $x(G - F_K^x) \geq 1$ is violated, which is a contradiction. \square

By Proposition 1,

$$\mathcal{P}_{G,K} = \text{conv} \left\{ x \in \{0,1\}^{|E|} : x(S) \geq 1, S \subseteq E, (G - S)_K \text{ is not two-connected} \right\}$$

and it is a set covering polytope. The following results immediately follow from properties of set covering polytopes.

Theorem 1. *Let $G = (V, E)$ be a graph and $K > 0$ a given constant.*

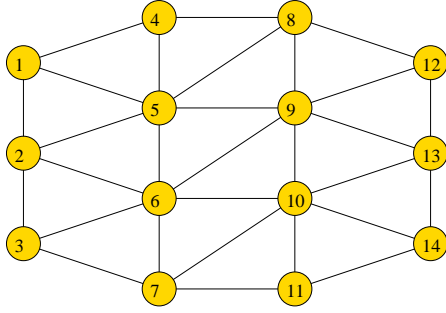


Fig. 1. An example for which cut constraints are not facet-defining ($K = 4$)

1. $\mathcal{P}_{G,K}$ is fully-dimensional if and only if $(G - e)_K$ is two-connected for each $e \in E$.

If $\mathcal{P}_{G,K}$ is fully-dimensional, then, for each $e \in E$,

2. $x_e \geq 0$ defines a facet of $\mathcal{P}_{G,K}$ if and only if $(G - \{e, f\})_K$ is two-connected for all $f \in E \setminus \{e\}$;
3. $x_e \leq 1$ defines a facet of $\mathcal{P}_{G,K}$.

Proof. Direct consequences of Proposition 2.1 in Sassano [14]. \square

In the remainder of this paper, we assume that $\mathcal{P}_{G,K}$ is fully-dimensional.

3. Cut constraints

Classical inequalities used to impose that a network is two-edge-connected are cut constraints. These constraints are widely used to formulate the traveling salesman problem — in this case, they are equivalent to subtour elimination constraints — or the minimum-cost two-connected network problem. Given a subset of nodes $W \subseteq V$, $\emptyset \neq W \neq V$, the cut constraint imposes that there are at least two edges leaving W , i.e.

$$x(\delta(W)) \geq 2. \quad (3)$$

We want to characterize which cut constraints define facets of $\mathcal{P}_{G,K}$. Stoer [15] (Theorem 5.4, page 51) gives necessary and sufficient conditions for these constraints to define facets of the polyhedron of the solutions of low-connectivity constrained network design problems.

Unfortunately, the conditions of Stoer's theorem are not sufficient when we add bounded ring constraints, as illustrated by the following example. Consider the graph G depicted in Figure 1 with unit edge lengths and a bound $K = 4$

on the ring lengths. The cut defined by $W = \{1, 2, 3, 4, 5, 6, 7\}$ induces the facet $x(\delta(W)) \geq 2$ of the polytope of two-connected subnetworks of G . To determine if this inequality defines a facet of $\mathcal{P}_{G,K}$, we consider the vectors of $\mathcal{P}_{G,K}$ that lie on the face, i.e. such that $x(\delta(W)) = 2$. The pairs of edges in $\delta(W)$ whose incidence vectors have that property are

$$\begin{aligned} &\{4, 8\} \text{ and } \{5, 9\}, \\ &\{6, 10\} \text{ and } \{5, 9\}, \\ &\{6, 10\} \text{ and } \{7, 11\}, \\ &\{5, 8\} \text{ and } \{6, 9\}, \\ &\{7, 10\} \text{ and } \{6, 9\}. \end{aligned}$$

We can easily conclude that all points in the face defined by $x(\delta(W)) = 2$ satisfy the following equations :

$$\begin{aligned} x_{\{4,8\}} + x_{\{6,10\}} + x_{\{5,8\}} + x_{\{7,10\}} &= 1, \\ x_{\{5,9\}} + x_{\{7,11\}} + x_{\{6,9\}} &= 1. \end{aligned}$$

Therefore, the dimension of the face is at most $m - 2$ and $x(\delta(W)) \geq 2$ does not define a facet.

From this example, we see that we need additional conditions to characterize which cut constraints are facet-defining. For any pair of edges $e, f \in \delta(W)$, it is useful to know if there exists a vector of $\mathcal{P}_{G,K}$ lying in the face $x(\delta(W)) = 2$ whose corresponding graph contains e and f . This is the case if and only if the incidence vector of

$$C_{e,f} := E(W) \cup E(V \setminus W) \cup \{e, f\}$$

belongs to $\mathcal{P}_{G,K}$, i.e. if $(C_{e,f})_K$ is two-connected. A useful tool to represent and analyze the vectors belonging to the face defined by a cut constraint is the *ring-cut graph* defined below.

Definition 1 (Ring-cut graph).

Let $G = (V, E)$ be a graph, $K > 0$ a given constant, and $W \subseteq V$ a subset of nodes, $\emptyset \neq W \neq V$.

The ring-cut graph $RCG_{W,K} := (\delta(W), RCE_{W,K})$ induced by W is the graph defined by associating one node to each edge in $\delta(W)$ and by the set of edges

$$RCE_{W,K} = \{\{e, f\} \subseteq \delta(W) : (C_{e,f})_K \text{ is two-connected}\}.$$

The ring-cut graph corresponding to the previous example is depicted in Figure 2. Note that this graph is bipartite, which implies that the cut constraint does not define a facet, as we show below (see Theorem 2).

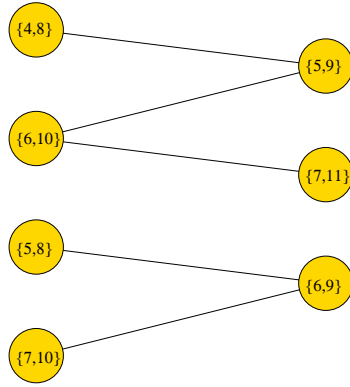


Fig. 2. Ring-cut graph for $W = \{1, 2, 3, 4, 5, 6, 7\}$ ($K = 4$)

With the help of the ring-cut graph, we are now ready to characterize which cut constraints are facet-defining. Before giving this characterization, the following technical lemmas give a necessary condition for an inequality to be facet-inducing and show that when this condition is satisfied, we can restrict our attention to the support of an inequality — variables having non-zero coefficients — in our facet proofs.

Lemma 1. *Let $a^T x \geq \alpha$ be a facet-defining inequality for $\mathcal{P}_{G,K}$. For each $f \in E$, if $a_f = 0$, there exists $\tilde{x} \in \{0, 1\}^{|E|}$ such that*

1. $\tilde{x}_f = 0$,
2. $F_K^{\tilde{x}}$ is two-connected,
3. $a^T \tilde{x} = \alpha$.

Proof. If no such \tilde{x} exists, then the face induced by $a^T x \geq \alpha$ is included in the facet defined by $x_f \leq 1$, and $a^T x \geq \alpha$ is not facet-defining. \square

Lemma 2. *Let $a^T x \geq \alpha$ and $b^T x \geq \beta$ be two valid inequalities for $\mathcal{P}_{G,K}$, such that the face induced by $a^T x = \alpha$ is included in the face induced by $b^T x = \beta$. If $a_f = 0$ for some $f \in E$ and if there exists $\tilde{x} \in \{0, 1\}^{|E|}$ satisfying the conditions of Lemma 1, then $b_f = 0$.*

Proof. If \tilde{x} satisfies the conditions of Lemma 1 and if $a_f = 0$, then $\tilde{x} + e_f$ belongs also to the face induced by $a^T x = \alpha$. Since this face is included in the face induced by $b^T x = \beta$, \tilde{x} and $\tilde{x} + e_f$ belong to the face induced by $b^T x = \beta$, implying that

$$b_f \tilde{x}_f = b_f (\tilde{x}_f + 1)$$

and thus $b_f = 0$. \square

Theorem 2. *Let $G = (V, E)$ be a graph, $K > 0$ a given constant, and $W \subseteq V$ a subset of nodes, $\emptyset \neq W \neq V$. The inequality*

$$x(\delta(W)) \geq 2$$

defines a facet of $\mathcal{P}_{G,K}$ if and only if

1. *for all $e \in \delta(W)$, there exists $f \in \delta(W)$ such that $(C_{e,f})_K$ is two-connected;*
2. *in each connected component of $RCG_{W,K}$, there exists a cycle of odd cardinality;*
3. *for all $e \in E(W) \cup E(V \setminus W)$, there exist $f, g \in \delta(W)$ such that $(C_{f,g} \setminus \{e\})_K$ is two-connected.*

Proof.

Necessity

If there exists $e \in \delta(W)$ such that for all $f \in \delta(W)$, $(C_{e,f})_K$ is not two-connected, then there is no feasible solution of 2CNBR, and therefore no point in $\mathcal{P}_{G,K}$, such that $x_e = 1$ and $x(\delta(W)) = 2$. The face induced by $x(\delta(W)) \geq 2$ is thus completely included in the face defined by $x_e \geq 0$, and the cut constraint does not define a facet.

Let $S \subseteq \delta(W)$ be a connected component of $RCG_{W,K}$. If there is no cycle of odd cardinality in the subgraph $RCG_{W,K}(S)$, then this subgraph is bipartite. Therefore, there exist two subsets S_1 and S_2 such that $S_1 \cup S_2 = S$, $S_1 \cap S_2 = \emptyset$ and $RCE_{W,K}(S_1) = RCE_{W,K}(S_2) = \emptyset$. Any graph whose incidence vector belongs to $\mathcal{P}_{G,K}$ and lies in the face $x(\delta(W)) = 2$ contains exactly two edges e and f in the cut. These edges are such that $(C_{e,f})_K$ is two-connected, there is thus one edge between e and f in $RCE_{W,K}$. Therefore, either one of these edges belongs to S_1 and the other to S_2 , or the two edges belong to $\delta(W) \setminus S$. It is easy to see that this implies that the incidence vector of the graph satisfies the following system of equalities :

$$\begin{aligned} 2x(S_1) + x(\delta(W) \setminus S) &= 2, \\ 2x(S_2) + x(\delta(W) \setminus S) &= 2. \end{aligned}$$

This means that all points in the face $x(\delta(W)) = 2$ satisfy a system of equalities of dimension 2, the dimension of the face is thus at most $m - 2$ and the cut constraint does not define a facet.

The necessity of the last condition is a direct consequence of Lemma 1.

Sufficiency

Let the conditions be satisfied for some inequality $a^T x := x(\delta(W)) \geq 2$. Let $b^T x \geq \beta$ be a facet-defining inequality such that the face F_a induced by $a^T x \geq 2$ in $\mathcal{P}_{G,K}$ is contained in the face F_b induced by $b^T x \geq \beta$. From Lemma 2 and the last condition, we already know that $b_e = 0$ for all $e \in E(W) \cup E(V \setminus W)$. The remainder of the proof is divided in two steps : we first show that the coefficients

of variables corresponding to edges in the same connected component of the ring-cut graph are equal in the inequalities $a^T x \geq 2$ and $b^T x \geq \beta$. Then we show that these coefficient are equal for edges belonging to different components.

1. $b_e = \gamma_S$ for all $e \in S$, for all connected component S of $RCG_{W,K}$.

Consider a tree rooted at k in S . In this tree, there is a unique path between k and any $l \in S$. We can partition S in two subsets S_1 and S_2 where

$$S_1 = \{l \in S : \text{the path from } k \text{ to } l \text{ contains an odd number of edges}\},$$

$$S_2 = \{l \in S : \text{the path from } k \text{ to } l \text{ contains an even number of edges}\}.$$

Since S contains a cycle of odd cardinality, either S_1 or S_2 contains at least two elements. Assume without loss of generality that $e, f \in S_1$, $e \neq f$. By the definition of S_1 , the unique path from e to f in the tree contains an even number of edges. Let $e = e_0, e_1, \dots, e_{2j} = f$ be this path. By the definition of the ring-cut graph, the existence of this path means that $(C_{e_i, e_{i+1}})_K$ and $(C_{e_{i+1}, e_{i+2}})_K$ are two-connected for $i = 0, \dots, 2(j-1)$. Therefore, $b_{e_i} = b_{e_{i+2}}$ and

$$b_e = b_{e_0} = b_{e_2} = \dots = b_{e_{2j}} = b_f.$$

Thus all x_e such that $e \in S_1$ have the same coefficient γ_{S_1} . The same reasoning applies to S_2 , and all x_e such that $e \in S_2$ have the same coefficient γ_{S_2} .

Since there exists a cycle of odd cardinality in S , there must exist an edge between two nodes in S_1 or between two nodes in S_2 . Without loss of generality, suppose there exists an edge between e and f in S_1 , i.e. $(C_{e,f})_K$ is two-connected. Moreover, since S is connected, there exists an edge between some $g \in S_1$ and some $h \in S_2$, and $(C_{g,h})_K$ is two-connected. Thus, $b_e + b_f = b_g + b_h$. Since $b_e = b_f = b_g = \gamma_{S_1}$ and $b_h = \gamma_{S_2}$, we can conclude that

$$\gamma_{S_1} = \gamma_{S_2} = \gamma_S$$

and all edges in S have the same coefficient.

2. $\gamma_S = \gamma_T$ for all connected components S, T of $RCG_{W,K}$.

To prove that all edges in $\delta(W)$ have the same coefficient in the inequality, it remains to prove that the variable coefficients are equal for two edges belonging to different connected components S and T of $RCG_{W,K}$. For all $e \in S$, there exists f such that $(C_{e,f})_K$ is two-connected. Therefore, $f \in S$ and $b_e + b_f = 2\gamma_S$. Similarly, we can find $g, h \in T$ such that $(C_{g,h})_K$ is two-connected and $b_g + b_h = 2\gamma_T$. So, $b_e + b_f = b_g + b_h$ and we can conclude that $\gamma_S = \gamma_T = \gamma$.

We have proved that $b = \gamma a$. Since F_b cannot define a facet if $b \leq 0$, we have $\gamma > 0$. Thus $x(\delta(W)) \geq 2$ and $b^T x \geq \beta$ define the same facet $F_a = F_b$. \square

4. Ring-cut inequalities

We saw in Section 3 that the ring-cut graph allows to determine if a cut constraint is facet-defining. In this section, we use the ring-cut graph to generate new valid inequalities for the 2CNBR problem.

Consider the graph depicted in Figure 3 with unit edge lengths and a bound $K = 4$ on the ring lengths. The ring-cut graph corresponding to the cut defined by $W = \{1, 2, 3, 4, 5, 6\}$ is depicted in Figure 4. Since this graph is bipartite, it follows from Theorem 2 that the cut constraint $x(\delta(W)) \geq 2$ does not define a facet. Our aim is to find some valid inequality that dominates the cut constraint. The cut constraint is not facet-defining because there are not enough edges in the ring-cut graph, i.e. not enough pairs of edges $e, f \in \delta(W)$ such that $(C_{e,f})_K$ is two-connected. A way to find a stronger inequality is to extend the search for feasible points to solutions using more than two edges in the cut.

To do so, we extend the notation $C_{e,f}$ to any subset $S \subseteq \delta(W)$ by defining

$$C_S := E(W) \cup E(V \setminus W) \cup S.$$

For the example of Figure 3, we can easily enumerate all the minimal subsets $S \subseteq \delta(W)$ such that $(C_S)_K$ is two-connected. These subsets are :

$$\begin{aligned} & \{\{3, 7\}, \{4, 8\}\}, \\ & \{\{5, 7\}, \{4, 8\}\}, \\ & \{\{5, 9\}, \{4, 8\}\}, \end{aligned}$$

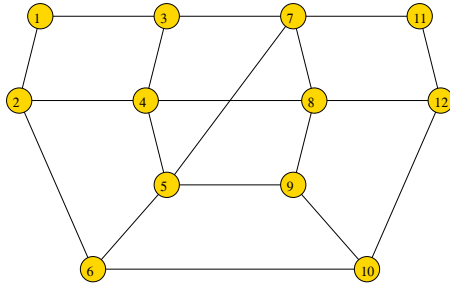


Fig. 3. Graph for which some ring-cut inequalities are valid

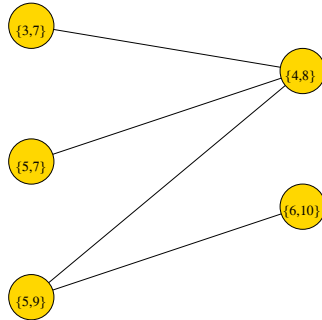


Fig. 4. The ring-cut graph for $W = \{1, 2, 3, 4, 5, 6\}$

$$\begin{aligned} & \{\{5, 9\}, \{6, 10\}\}, \\ & \{\{3, 7\}, \{5, 7\}, \{5, 9\}\}. \end{aligned}$$

It is easy to see that the incidence vectors of these subsets satisfy

$$x_{\{3,7\}} + x_{\{5,7\}} + x_{\{5,9\}} + 2x_{\{4,8\}} + 2x_{\{6,10\}} = 3.$$

Moreover, a subset $\bar{S} \subseteq \delta(W)$ induces a two-connected $(C_{\bar{S}})_K$ if and only if \bar{S} contains one of the above subsets. Therefore, any feasible solution satisfies the inequality

$$x_{\{3,7\}} + x_{\{5,7\}} + x_{\{5,9\}} + 2x_{\{4,8\}} + 2x_{\{6,10\}} \geq 3.$$

The next proposition generalizes this result.

Proposition 2. *Let $G = (V, E)$ be a graph, $K > 0$ a given constant, $W \subseteq V$ a subset of nodes, $\emptyset \neq W \neq V$.*

If $S \subseteq \delta(W)$ is an independent subset in the ring-cut graph $RCG_{W,K}$, then

$$x(S) + 2x(\delta(W) \setminus S) \geq 3 \tag{4}$$

is a valid inequality for the 2CNBR problem.

Proof. Let F be a feasible solution to the 2CNBR problem and x be its incidence vector. Since F is two-connected, $x(\delta(W)) \geq 2$ and therefore,

$$x(S) + 2x(\delta(W) \setminus S) \geq 2.$$

If this inequality is satisfied as equality by F , then F contains exactly two edges e and f in $\delta(W)$. These edges belong to S since edges in S are the only edges having a coefficient equal to 1 in the inequality. Moreover, $(C_{e,f})_K$ is two-connected since it includes F , thus there is an edge between e and f in the ring-cut graph. Since S is an independent subset, this leads to a contradiction. We can conclude that (4) is a valid inequality for the 2CNBR problem. \square

Inequalities (4) are called *ring-cut inequalities*. Note that if the ring-cut graph is bipartite, as is the case in Figure 4, then it is possible to partition $\delta(W)$ in two independent subgraphs S_1 and S_2 that define the two ring-cut inequalities

$$\begin{aligned} x(S_1) + 2x(S_2) &\geq 3, \\ 2x(S_1) + x(S_2) &\geq 3. \end{aligned}$$

The sum of these inequalities, divided by 3, gives the cut constraint $x(\delta(W)) \geq 2$. Therefore, in this case, ring-cut inequalities are stronger than cut constraints.

We were not able to characterize completely which ring-cut inequalities are facet-defining. However, we present here a set of necessary conditions, as well as two sets of sufficient conditions.

Theorem 3. Let $G = (V, E)$ be a graph, $K > 0$ a given constant, $W \subseteq V$ a subset of nodes, $\emptyset \neq W \neq V$.

If $S \subseteq \delta(W)$ is an independent subset in the ring-cut graph $RCG_{W,K}$ and if

$$x(S) + 2x(\delta(W) \setminus S) \geq 3$$

defines a facet of $\mathcal{P}_{G,K}$, then

1. for all $e \in S$, either there exists $f \in \delta(W) \setminus S$ such that $(C_{e,f})_K$ is two-connected or there exist $g, h \in S$ such that $(C_{\{e,g,h\}})_K$ is two-connected;
2. for all $e \in \delta(W) \setminus S$, there exists $f \in S$ such that $(C_{e,f})_K$ is two-connected;
3. there exist $e, f, g \in S$ such that $(C_{\{e,f,g\}})_K$ is two-connected.

Proof. First, note that the incidence vector of a graph lies in the face defined by (4) if and only if the graph contains three edges belonging to S and no edge in $\delta(W) \setminus S$, or one edge in S and one edge in $\delta(W) \setminus S$.

Suppose one of the first two conditions is not satisfied for some $e \in \delta(W)$. It is easy to see from the remark above that in this case, there is no vector of $\mathcal{P}_{G,K}$ with $x_e = 1$ lying in the face defined by (4). Therefore, this face is completely included in the face defined by $x_e \geq 0$ and (4) does not define a facet.

If the third condition is not satisfied, then any graph whose incidence vector lies in the face defined by (4) contains exactly one edge in S and one edge in $\delta(W) \setminus S$. Therefore, these vectors satisfy the following system of equalities :

$$\begin{aligned} x(S) &= 1, \\ x(\delta(W) \setminus S) &= 1. \end{aligned}$$

This means that all points in the face defined by (4) satisfy a system of equalities of dimension 2, thus the dimension of the face is at most $m - 2$ and (4) does not define a facet. \square

Consider again the example of Section 3. The ring-cut graph depicted in Figure 2 is bipartite, and

$$x_{\{4,8\}} + x_{\{6,10\}} + x_{\{5,8\}} + x_{\{7,10\}} + 2x_{\{5,9\}} + 2x_{\{7,11\}} + 2x_{\{6,9\}} \geq 3$$

is a valid ring-cut inequality. However, it is easy to check that the third condition of Theorem 3 is not satisfied, and therefore the inequality is not facet-defining.

Theorem 4. Let $G = (V, E)$ be a graph, $K > 0$ a given constant, $W \subseteq V$ a subset of nodes, $\emptyset \neq W \neq V$.

If $S \subseteq \delta(W)$ is an independent subset in the ring-cut graph $RCG_{W,K}$, and if

1. the bipartite subgraph $(\delta(W), \delta_{RCG_{W,K}}(S))$ of $RCG_{W,K}$ — obtained by considering only edges leaving S — is connected;
2. there exist $e, f, g \in S$ such that $(C_{\{e,f,g\}})_K$ is two-connected;

3. for all $e \in E(W) \cup E(V \setminus W)$, there exist $f \in S$ and $g \in \delta(W) \setminus S$ such that $(C_{f,g} \setminus \{e\})_K$ is two-connected;

then

$$x(S) + 2x(\delta(W) \setminus S) \geq 3$$

defines a facet of $\mathcal{P}_{G,K}$.

Proof. Let the conditions be satisfied for some inequality $a^T x := x(S) + 2x(\delta(W) \setminus S) \geq 3$. Let $b^T x \geq \beta$ be a facet-defining inequality such that the face F_a induced by $a^T x \geq 3$ in $\mathcal{P}_{G,K}$ is contained in the face F_b induced by $b^T x \geq \beta$. From Lemma 2 and the last condition, we already know that $b_e = 0$ for all $e \in E(W) \cup E(V \setminus W)$. The remaining of the proof is divided in three steps : we first show that the coefficients of variables corresponding to edges belonging to S are equal in the inequalities $a^T x \geq 3$ and $b^T x \geq \beta$. A similar proof leads to the same result for edges in $\delta(W) \setminus S$, then we establish the relationship between all the coefficients.

1. $b_e = \gamma$ for all $e \in S$.

We first show that b_e has the same value γ for all edges belonging to S . Consider two edges $e, f \in S$, $e \neq f$. Since $(\delta(W), \delta_{RCG_{W,K}}(S))$ is connected and bipartite, there exists a path of even cardinality from e to f in $RCG_{W,K}$. By arguments similar to those of Theorem 2, we can conclude that $b_e = b_f$.

2. $b_e = \bar{\gamma}$ for all $e \in \delta(W) \setminus S$.

A similar proof leads to the fact that b_e has the same value $\bar{\gamma}$ for all edges belonging to $\delta(W) \setminus S$.

3. $\bar{\gamma} = 2\gamma$.

There exist $e, f, g \in S$ such that $(C_{\{e,f,g\}})_K$ is two-connected. Moreover, since $(\delta(W), \delta_{RCG_{W,K}}(S))$ is connected, there exists $h \in \delta(W) \setminus S$ such that $(C_{e,h})_K$ is two-connected. Therefore, $b_e + b_f + b_g = b_e + b_h$. Since $b_e = b_f = b_g = \gamma$ and $b_h = \bar{\gamma}$, we can conclude that $\bar{\gamma} = 2\gamma$.

We have proved that $b = \gamma a$. Since F_b cannot define a facet if $b \leq 0$, we have $\gamma > 0$. Thus $x(S) + 2x(\delta(W) \setminus S) \geq 3$ and $b^T x \geq \beta$ define the same facet $F_a = F_b$. \square

The example of ring-cut inequality given at the beginning of this section satisfies these conditions, and therefore is facet-defining.

5. Node-cut and subset inequalities

After examining inequalities that impose two-edge-connectivity, we study in this section node-cut constraints that impose two-node-connectivity. Given a node $z \in V$ and a subset of nodes $W \subset V \setminus \{z\}$, $\emptyset \neq W \neq V \setminus \{z\}$, the cut induced by

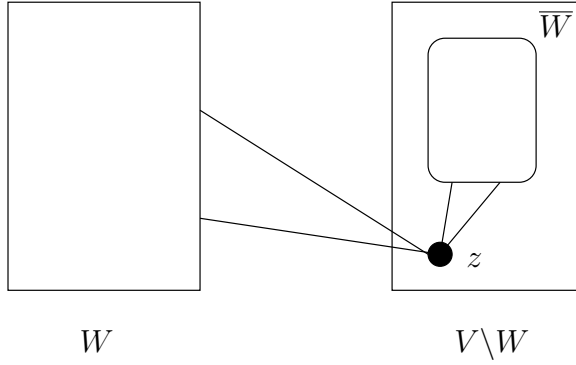


Fig. 5. $G - (\delta(W) \setminus \delta(\{z\}))$

W in $G - z$ must contain at least one edge, otherwise z is an articulation point. The node-cut constraint is thus

$$x(\delta_{G-z}(W)) \geq 1. \quad (5)$$

If S is the subset of edges not incident to z in the cut, i.e. $S = \delta_{G-z}(W) = \delta(W) \setminus \delta(\{z\})$, it is clear that $G - S$ is not two-connected since z is an articulation point in this graph, as illustrated in Figure 5. Therefore, S defines a subset inequality which is the node-cut constraint $x(\delta_{G-z}(W)) \geq 1$, and all node-cut constraints are in fact subset inequalities.

The next theorem shows under which conditions subset inequalities define facets of $\mathcal{P}_{G,K}$.

Theorem 5. *Let $G = (V, E)$ be a graph, $K > 0$ a given constant, and $S \subseteq E$ a subset of edges such that $(G - S)_K$ is not two-connected. Then*

$$x(S) \geq 1$$

defines a facet of $\mathcal{P}_{G,K}$ if and only if

1. *for all $e \in S$, $(G - (S \setminus \{e\}))_K$ is two-connected;*
2. *for all $f \in E \setminus S$, there exists $e \in S$ such that $G - (S \cup \{f\} \setminus \{e\})_K$ is two-connected.*

Proof. Direct consequence of Theorem 2.1 in Sassano [14]. □

Consider again the graph depicted in Figure 1, with $K = 4$. The node-cut constraint defined by $W = \{1, 2, 3, 4, 5, 6, 7\}$ and $z = 8$,

$$x_{\{5,9\}} + x_{\{6,9\}} + x_{\{6,10\}} + x_{\{7,10\}} + x_{\{7,11\}} \geq 1,$$

does not define a facet since it is induced by

$$S = \{\{5, 9\}, \{6, 9\}, \{6, 10\}, \{7, 10\}, \{7, 11\}\}$$

and $(G - (S \setminus \{\{7, 11\}\}))_4$ is not two-connected. Therefore,

$$S = \{\{5, 9\}, \{6, 9\}, \{6, 10\}, \{7, 10\}\}$$

induces a stronger subset inequality,

$$x_{\{5,9\}} + x_{\{6,9\}} + x_{\{6,10\}} + x_{\{7,10\}} \geq 1.$$

This inequality is facet-defining since removing any edge $e \in S$ from S induces a two-connected $(G - (S \setminus \{e\}))_K$.

This shows that subset inequalities are a useful generalization of node-cut inequalities for the 2CNBR problem. Note that subset inequalities are also valid for two-connected networks (without ring constraints). However, in this case, the only interesting subset inequalities (i.e. those that are facet-defining) are node-cut constraints, as shown in the next proposition.

Proposition 3. *Let $G = (V, E)$ be a graph, and $S \subseteq E$ a subset of edges such that $G - S$ is not two-connected.*

If $x(S) \geq 1$ defines a facet of $\mathcal{P}_{G,\infty}$ then this inequality is a node-cut constraint.

Proof. Since $G - S$ is not two-connected, it contains at least one articulation point $z \in V$, or $G - S$ is not connected, in which case z can be any node in V for the proof to hold. Let $W \subseteq V$ be a subset of nodes defining a connected component of $(G - S) - z$. Since $x(S) \geq 1$ defines a facet, by Theorem 5, for any $e \in S$, $G - (S \setminus \{e\})$ is two-connected. Therefore, at least one edge f of $\delta(W) \setminus \delta(\{z\})$ belongs to $G - (S \setminus \{e\})$. The only possible choice for f is $f = e$, thus $e \in \delta(W) \setminus \delta(\{z\})$. We can conclude that $S \subseteq \delta(W) \setminus \delta(\{z\})$.

If the two sets are equal, then

$$S = \delta_{G-z}(W)$$

and the subset inequality is a node-cut constraint.

Otherwise, there exists some edge $e \in \delta(W) \setminus \delta(z)$ such that $e \notin S$, i.e. $e \in G - S$, which is in contradiction with the fact that W is a connected component of $(G - S) - z$. \square

6. Node-partition inequalities

The polyhedron of connected networks is completely described by *partition inequalities* (Grötschel and Monma [9]). Given a partition W_1, W_2, \dots, W_p ($p \geq 2$) of V into p nonempty subsets, the corresponding partition inequality is

$$\frac{1}{2} \sum_{i=1}^p x(\delta(W_i)) \geq p - 1.$$

Since a two-connected network remains connected when a node is removed,

$$\frac{1}{2} \sum_{i=1}^p x(\delta_{G-z}(W_i)) \geq p - 1$$

is a valid inequality for the polyhedron of two-connected networks, and therefore for 2CNBR, where W_1, W_2, \dots, W_p ($p \geq 2$) is a partition of $V \setminus \{z\}$. These inequalities are called *node-partition inequalities* (Grötschel and Monma [9]).

The following theorem gives sufficient conditions for node-partition inequalities to define facets of $\mathcal{P}_{G,K}$.

Theorem 6. *Let $G = (V, E)$ be a graph, $K > 0$ a given constant, $z \in V$ a node of the graph and W_1, W_2, \dots, W_p ($p \geq 2$) a partition of $V \setminus \{z\}$. Let \hat{G} denote the graph $(G_K - z)/W_1/\dots/W_p$. If*

1. \hat{G} is two-connected;
2. $(G(W_i \cup \{z\}) - e)_K$ is two-connected for all $e \in G(W_i \cup \{z\})$;
3. For all $e := \{i, j\}$ with $i \in W_i$ and $j \in W_j$, $i \neq j$,
 - $\{z, i, j\}$ form a feasible cycle;
 - for all $f \in \delta(\{z\})$, there exists a feasible cycle C using z and e , not using f , such that $|C \cap \delta(W_k)| \leq 2$ for $k = 1, \dots, p$.

then

$$a^T x := \frac{1}{2} \sum_{i=1}^p x(\delta_{G-z}(W_i)) \geq p - 1$$

defines a facet of $\mathcal{P}_{G,K}$.

Proof. Let the conditions be satisfied. We show that there are $|E|$ affinely independent vectors in the face defined by the node-partition inequality.

Let E' be the set of all edges whose coefficients in $a^T x \geq p - 1$ are equal to zero, i.e.

$$E' = \bigcup_{i=1}^p E(W_i) \cup \delta(z).$$

By the first condition, the graph $\hat{G} = (\hat{V}, \hat{E})$ contains $|\hat{E}|$ spanning trees whose incidence vectors are affinely independent (see Theorem 4.10 in [9]). Any such tree of \hat{G} can be augmented by E'_K to a set $C \subseteq E$ whose incidence vector belongs to $\mathcal{P}_{G,K}$. To prove this, we will show that $C = C_K$. By Theorem 6.3 in [11], the first two conditions imply that C is two-connected, and therefore the incidence vector of $C = C_K$ belongs to $\mathcal{P}_{G,K}$.

Consider an edge $e := \{i, j\} \in C$ arising from the tree in \hat{G} . By the third condition, $\{z, i, j\}$ form a feasible cycle and $e \in C_K$. If $e \in E'_K$, then $e \in C_K$ by the second condition. Therefore, $C \subseteq C_K$. The reverse inclusion is obvious.

We have thus constructed $|\hat{E}|$ affinely independent vectors lying in the face $a^T x \geq p - 1$.

Now consider an edge $f \in E'$. If there exists some vector in the $|\hat{E}|$ vector constructed before such that $x_f = 0$, then setting $x_f = 1$ in this vector leads to a new vector affinely independent with the others. If $x_f = 1$ for all the vectors in the family above, the same proof as before applied to $C \setminus \{f\}$ instead of C implies that the incidence vector of $C \setminus \{f\}$ belongs to $\mathcal{P}_{G,K}$, for any C of the family. We have now $|\hat{E}| + |E'| = |E| - |\delta(\{z\})|$ affinely independent vectors lying in the face.

Consider $e := \{w_i, z\} \in \delta(\{z\})$, with $w_i \in W_i$. We construct a tree \hat{T} in \hat{G} , corresponding to a subset of edges T in E , such that $(T \cup E' \setminus \{e\})_K$ is two-connected, leading to $|\delta(\{z\})|$ new affinely independent vectors lying in the face.

Let $f \in \delta_{G-z}(W_i)$ be an edge incident to w_i . By the last condition, there exists a feasible cycle using f and z but not e . Let T be the set of edges in this cycle that belong to $\bigcup_{i=1}^p \delta_{G-z}(W_i)$. The corresponding edges in \hat{T} form a path in \hat{G} , as each W_i is crossed at most once by the cycle by the last part of the condition. Since T was constructed from a feasible cycle, we have $T \cup E' \setminus \{e\} = (T \cup E' \setminus \{e\})_K$.

If \hat{T} is connected, it is a tree and the construction is finished. Otherwise, let W be the set of nodes in the connected component defined by \hat{T} . Since \hat{G} is two-connected, there is at least one edge leaving a node \hat{w} different from the node corresponding to W_i . Let f be the corresponding edge in G . Adding f to T increases the size of the subtree \hat{T} . It is clear that f is not incident to w_i , and therefore, by the third condition, f belongs to a feasible cycle in $T \cup E' \setminus \{e\}$ and we preserve the fact that $T \cup E' \setminus \{e\} = (T \cup E' \setminus \{e\})_K$. This construction is repeated until \hat{T} forms a tree spanning \hat{G} .

We have thus constructed a tree in \hat{G} , such that $T \cup E' \setminus \{e\} = (T \cup E' \setminus \{e\})_K$. Again, by Theorem 6.3 in [11], $T \cup E' \setminus \{e\}$ is two-connected and therefore, its incidence vector belongs to $\mathcal{P}_{G,K}$. \square

7. Algorithmic aspects

In this section, we study the algorithmic aspects of the problem: we first address the problem of checking the feasibility of a network, then we consider the separation problem associated to each class of valid inequalities. We finish this section with a description of the Branch-and-Cut algorithm in which these inequalities were embedded.

7.1. Checking feasibility

An important issue for implementing efficient algorithms for the 2CNBR problem is feasibility checking. A simple (but inefficient) way to evaluate whether a graph G is feasible is to check if it is two-connected, and if each edge belongs to a feasible ring (by computing a shortest path not using the edge, between its endpoints). In other words, it is the same as checking that $G = G_K$ and that G_K is two-connected.

In itself, checking if the second condition is satisfied is an important issue, e.g. in the separation of subset and ring-cut inequalities (see Sections 7.4 and 7.3). Moreover, we are interested in feasible graphs of minimum cost. If $G \neq G_K$, the cost of G_K is less than the cost of G . Therefore, if G_K is two-connected, it provides a feasible solution of cost less than the cost of G .

Instead of checking feasibility of G , we are thus more interested in the feasibility of G_K , which is simply a two-connectivity test for G_K . This test can be performed by an $O(m)$ depth-first search algorithm (see e.g. Aho et al. [1]). The bottleneck in this approach is that we still have to construct G_K , which requires to check for each edge if it belongs to a feasible ring. Instead of doing this check before testing two-connectivity, we propose to adapt the depth-first search algorithm in order to reduce the number of searches for feasible cycles.

A depth-first search of an undirected graph $G = (V, E)$ partitions the edges in E into two sets T and B . An edge $\{v, w\}$ is placed in T if node w has not been previously visited when we are at node v considering edge $\{v, w\}$. Otherwise, edge $\{v, w\}$ is placed in B . The edges in T are called *tree edges*, and those in B *back edges*. Nodes in $|V|$ are numbered in the order they are visited during the depth-first search, $\text{num}(v)$ denoting the number of v . If v was visited before w and if $\{v, w\} \in T$, v is called the *parent* of w and w is a *child* of v .

To check two-connectivity of a graph, we define for each node $v \in V$

$$\begin{aligned} \text{inf}(v) := \min (\{ \text{num}(v) \} \cup \{ \text{num}(w) : \{v, w\} \in B \} \\ \cup \{ \text{inf}(w) : w \text{ is a child of } v \}). \end{aligned}$$

If a node v is not the root of the depth-first search, then it can be shown that v is an articulation point if and only if it has a child w such that $\text{inf}(w) \geq \text{num}(v)$. If v is the root of the depth-first search, then v is an articulation point if we backtrack to v before having visited all nodes (see Aho et al. [1] for details). Moreover, the computation of $\text{inf}(v)$ for all nodes can be embedded in the depth-first search algorithm, leading to a computing time of $O(m)$ to check two-connectivity of a graph.

We now turn back to the problem of checking the two-connectivity of G_K . If we apply the two-connectivity algorithm to G , edges that do not belong to any feasible cycle must be ignored in the search, since they do not belong to G_K . Moreover, if a back edge $\{v, w\} \in B$, examined from node v , is such that

$\text{num}(w) \geq \text{inf}(v)$, then v and w are in the same two-connected component even if the edge $\{v, w\}$ is removed from the graph. Therefore, this edge is not necessary to ensure two-connectivity and we do not need to check if it belongs to G_K , reducing the number of necessary checks.

To check if $e \in G_K$, we compute the shortest cycle using e in G . If the length of this cycle is less than or equal to K , all the edges belonging to the cycle are in G_K , and we can keep this information in order to avoid redundant checking. This is done by giving a mark to each edge which is set to true as soon as we have found a feasible cycle containing the edge. Moreover, the cycle may use a node v such that $\text{num}(v) < \text{inf}(w)$ for some w in the cycle, allowing us to decrease $\text{inf}(w)$ faster than allowed by the classical depth-first search approach, thus again reducing the amount of work needed.

Computing the shortest cycle to which an edge belongs requires the computation of a shortest path between the endpoints of the edge, and has thus a complexity of $O(n^2)$ if we use Dijkstra's algorithm. This subroutine is called at most m times, leading to a complexity of $O(mn^2)$ for feasibility checking. Even if this worst-case complexity is the same as the simple approach described at the beginning of this section, it is more efficient in practice since the number of shortest cycle computations is usually far less than m , while it would always be equal to m with the straightforward approach.

7.2. Separation of cut constraints

The separation of cut constraints $x(\delta(W)) \geq 2$ can be carried out by computing a minimum cut in the graph, with capacities given by the current LP solution. This can be done in polynomial time, e.g. by the Gomory-Hu algorithm [8] that requires $n - 1$ maximum flow computations.

7.3. Separation of ring-cut inequalities

The ring-cut inequality associated with an independent subset $S \subseteq \delta(W)$ in the ring-cut graph $RCG_{W,K}$ defined by a subset of nodes W is

$$x(S) + 2x(\delta(W) \setminus S) \geq 3.$$

The separation of ring-cut inequalities is performed using a greedy heuristic, since the separation problem is NP-hard. A proof of the NP-hardness of the separation problem is provided in appendix. As we use the Gomory-Hu algorithm to separate cut constraints, we benefit from this information in the separation of ring-cut inequalities, considering the $n - 1$ minimum cuts provided by the Gomory-Hu tree as good candidates. To each cut of capacity less than 3 in the tree, we apply the greedy heuristic described in Algorithm 1 to determine an independent subset in the ring-cut graph corresponding to the cut.

Algorithm 1 Separation of ring-cut inequalities

Data : a graph $G = (V, E)$, a real number $K > 0$, a vector $\tilde{x} := (\tilde{x}_f)_{f \in E}$, a subset $W \subseteq V$, $\emptyset \neq W \neq V$ such that $\tilde{x}(\delta(W)) < 3$.

```

1:  $v := \tilde{x}(\delta(W))$ 
2:  $S := \emptyset$ 
3: for all  $e \in \delta(W)$  do
4:    $mark(e) := \text{false}$ 
5: while  $(v < 3)$  and  $(\exists e \in \delta(W) : mark(e) = \text{false})$  do
6:    $e := \arg \max\{\tilde{x}_f : f \in \delta(W), mark(f) = \text{false}\}$ 
7:   for all  $f \in \delta(W) : mark(f) = \text{false}$  do
8:     if  $(C_{e,f})_K$  is two-connected then
9:       {edge  $\{e, f\}$  belongs to the ring-cut graph}
10:       $v := v + \tilde{x}_f$ 
11:       $mark(f) := \text{true}$ 
12:    $mark(e) := \text{true}$ 
13:    $S := S \cup \{e\}$ 
14: if  $v < 3$  then
15:    $\{x(S) + 2x(\delta(W) \setminus S) \geq 3$  is a violated ring-cut inequality}

```

7.4. Separation of subset inequalities

We now study subset inequalities (1) that were used to formulate the 2CNBR problem. The subset inequality induced by a subset of edges $S \subseteq E$, such that $(G - S)_K$ is not two-connected, is $x(S) \geq 1$.

The separation of subset inequalities for the two-connected network problem ($K = \infty$) can be performed in polynomial time as facet-defining subset inequalities are node-cut constraints (by Proposition 3).

However, the general case is harder to solve. We want to find a subset $S \subseteq E$ such that $x(S) < 1$ and $(G - S)_K$ is not two-connected. Such a subset S does not necessarily define a node-cut and has no known special structure. The NP-completeness of this problem remains an open question, and we rely on the greedy heuristic described in Algorithm 2 to solve the separation problem.

The algorithm is divided in two phases : the first phase (lines 1-10) tries to find in a greedy way a subset $F \subseteq E$ such that $\tilde{x}(E \setminus F) < 1$ and F_K is not two-connected. If it succeeds, $x(E \setminus F) \geq 1$ is a violated subset inequality. The second phase (lines 11-18) is used to find $S \subseteq E \setminus F$ such that for all $e \in S$, $(G - (S \setminus \{e\}))_K$ is two-connected. The generated subset inequality is facet-defining by Theorem 5.

The main drawback of the greedy heuristic is that it often fails to find a suitable F . It is easy to adapt the algorithm to perform the exact separation by a backtracking procedure, enumerating all the subsets F such that $\tilde{x}(E \setminus F) < 1$. However, there is an exponential number of such subsets, and complete enumeration is not efficient in practice. An intermediate approach is to allow a fixed number of backtracking steps. Our computational experiments show that performing 10 backtracking steps is a good tradeoff between computing time and quality of the separation.

Algorithm 2 Separation of subset inequalities**Data :** a graph $G = (V, E)$, a real number $K > 0$, a vector $\tilde{\mathbf{x}} := (\tilde{x}_f)_{f \in E}$.

```

1: let  $F := \{e \in E : \tilde{x}_f > 0\}$ 
2: if  $F_K$  is two-connected then
3:   sort edges of  $F$  in increasing order of  $\tilde{x}_f$ ,
   such that  $F = \{j(1), \dots, j(|F|)\}$ ,  $\tilde{x}_{j(i)} \leq \tilde{x}_{j(k)}$  for all  $i < k$ .
4:    $i := 1$ 
5:    $v := 0$ 
6:   while  $v + \tilde{x}_{j(i)} < 1$  and  $i \leq |F|$  and  $F_K$  is two-connected do
7:      $F := F \setminus \{j(i)\}$ 
8:     if  $F_K$  is two-connected then
9:        $v := v + \tilde{x}_{j(i)}$ 
10:       $i := i + 1$ 
11: if  $F_K$  is not two-connected then
12:    $S := \emptyset$ 
13:   for all  $e \in E \setminus F$  do
14:     if  $(F \cup \{e\})_K$  is two-connected then
15:        $S := S \cup \{e\}$ 
16:     else
17:        $F := F \cup \{e\}$ 
18:        $\{x(S) \geq 1$  is a violated facetdefining subset inequality}
19: else
20:    $\{\text{no violated subset was inequality found}\}$ 

```

Another way of deriving violated subset inequalities comes from the separation of metric inequalities. Metric inequalities were introduced by Fortz et al. [6], and are valid for the original 2CNBR problem, but not for the set covering formulation studied here. Consider an edge $e := \{i, j\} \in E$ and a set of node potentials $(\alpha_k)_{k \in V}$, derived from a directed flow from i to j , and satisfying

$$\alpha_i - \alpha_j > K - d(i, j).$$

Then

$$\sum_{f \in E \setminus \{e\}} v_f x_f \geq x_e \quad (6)$$

is a valid inequality for 2CNBR (*metric inequality*) where

$$v_f = \min \left(1, \max \left(0, \frac{|\alpha_l - \alpha_k| - d(k, l)}{\alpha_i - \alpha_j + d(i, j) - K} \right) \right) \quad (7)$$

for all $f := \{k, l\} \in E \setminus \{e\}$. A heuristic for the separation of these inequalities was proposed in [6].

Let $S := \{f \in E \setminus \{e\} : v_f > 0\}$. If $(G - S)_K$ is not two-connected and $x(S) < 1$, then S defines a violated subset inequality. This situation is often met in practice. Moreover, if $v_f = 1$ for all $f \in S$, then the subset inequality is stronger than the metric inequality.

7.5. Separation of node-partition inequalities

Given a node $z \in V$ and a partition W_1, W_2, \dots, W_p ($p \geq 2$) of $V \setminus \{z\}$ into p nonempty subsets, the corresponding node-partition inequality is

$$\frac{1}{2} \sum_{i=1}^p x(\delta(W_i)) \geq p - 1.$$

In Fortz et al. [6], these inequalities are separated using a heuristic suggested by Grötschel et al. [10]. However, it is possible to perform the exact separation in polynomial time. The first separation algorithm for these inequalities was given by Cunningham [3] and requires $|E|$ min-cut computations. Barahona [2] reduced this computing time to $|V|$ min-cut computations.

Separating node-partition inequalities can thus be done for each node $z \in V$ by applying Barahona's algorithm to $G - z$. This requires $|V|^2$ min-cut computations.

In order to reduce this number, note that if G is a two-edge-connected network, any articulation point in G has a degree at least equal to 4. Therefore, we decided to apply Barahona's algorithm to $G - z$ only if $x(\delta(z)) > 3$, which leads to a much faster separation procedure.

7.6. Separation of weighted partition inequalities

Another class of valid inequalities arises from a partition of the node set (Fortz et al. [6]). Given a partition W_1, W_2, \dots, W_p ($p \geq 2$) of V ,

$$\frac{1}{2} \sum_{i=1}^p \sum_{e \in \delta(W_i)} (K - d_e) x_e \geq (p - 1)K \quad (8)$$

is a valid inequality for 2CNBR, called *weighted partition inequality*.

Again, in Fortz et al. [6], these inequalities are separated heuristically. But the separation of weighted partition inequalities can also be performed using Barahona's algorithm, as the weighted partition inequality (8) can be rewritten

$$\frac{1}{2} \sum_{i=1}^p \sum_{e \in \delta(W_i)} \frac{K - d_e}{K} x_e \geq p - 1.$$

Solving the separation problem for partition inequalities for the vector $\tilde{\mathbf{x}}$ defined by $\tilde{x}_e := \frac{K - d_e}{K} x_e$, $e \in E$ is thus equivalent to solving the separation problem for weighted partition inequalities for x .

7.7. Implementation of the Branch-and-Cut algorithm

We now describe some strategic choices that were made in the implementation of our Branch-and-Cut algorithm for the 2CNBR problem. Our aim here is not to describe in detail the general Branch-and-Cut framework, but to emphasize the problem-specific aspects of our algorithm. A more complete description of the algorithm can be found in Fortz [5].

In our numerical experiments, we suppose for all instances that the original graph is complete, and therefore, the problem has $\frac{n(n-1)}{2}$ variables. However, it is possible that some of these edges have such a large length that they cannot belong to a feasible cycle. Such edges are removed in the preprocessing step, allowing a substantial decrease in the number of edges for instances where edge lengths are Euclidean.

Since we suppose edge lengths satisfy the triangle inequality, this preprocessing can be done in $O(n^3)$, checking for each triplet (i, j, k) of nodes if $d(i, j) + d(j, k) + d(k, i) \leq K$ and keeping only edges that belong to such a feasible triangle.

The initial linear program is defined by degree constraints $x(\delta(v)) \geq 2$ for all $v \in V$.

An important issue in the effectiveness of a Branch-and-Cut algorithm is the computation of good upper bounds. We developed an effective Tabu Search heuristic for this problem (Fortz [5]). In parallel with the Branch-and-Cut algorithm, 600 iterations of this heuristic are performed in order to obtain a good upper bound.

Moreover, we try to transform each LP-solution obtained in the Branch-and-Cut to a feasible solution by rounding up to 1 all the variables with fractional value.

The pool used to store generated inequalities is the standard pool in ABA-CUS (Thienel [16]). We start with a pool size equal to 100 times the number of nodes in the network, and we allow this size to be increased dynamically if necessary. All the generated inequalities are put in the pool and are dynamic, i.e. they are removed from the current LP when they are not active. The separation of valid inequalities is performed as follows. We first separate inequalities from the standard pool. If all the inequalities in the pool are satisfied by the current LP-solution, we separate the classes of inequalities introduced in the previous sections in the following order :

1. weighted partition inequalities;
2. subset inequalities;
3. cut constraints;
4. ring-cut inequalities;
5. node-partition inequalities;
6. metric inequalities.

This order was chosen after a series of numerical experiments, the choice of inequalities separated first seemingly being the best trade-off between separation time and efficiency of the cuts.

Moreover, we go to the next class of inequalities only if the number of generated cutting planes is less than 50. Otherwise, we solve the LP again and restart the separation procedure.

All inequalities are global (i.e. valid in the whole tree), except ring-cut inequalities that are valid locally. This is due to the fact that setting a variable to 0 may change the ring-cut graph generated by a cut not containing the corresponding edge.

8. Computational Results

We present in this section numerical results obtained for the 2CNBR problem with our Branch-and-Cut algorithm. This algorithm was implemented using ABACUS 2.0 and CPLEX 4.0, and tested on a SUN Sparc Ultra 1 workstation with a 166 Mhz processor and 128 Mo RAM. We fixed the maximum CPU time to 10 hours, except for randomly generated problems with 40 and 50 nodes, where it was limited to 3 hours, due to the large number of problems to solve. Moreover, for these large problems, we noticed that the bounds did not improve much after 3 hours.

Results are presented here for instances where edge lengths and costs are equal to the rounded Euclidean distances. Tests were made for different values of the bound, for instances coming from real applications, with 12, 17, 30 and 52 nodes, and for random problems with nodes uniformly generated in a square of size 250×250 . "Real" instances come from the network of Belgacom (52 nodes) and subsets of these nodes. Random problems with 10 to 50 nodes were generated, and we tested five instances of each size.

$ V $	number of nodes in the graph
$ E $	number of edges after preprocessing
K	bound on the length of cycles
p/o	for random problems, total number of problems/number of problems solved to optimality
# ineq.	number of inequalities generated
# B&B nodes	number of Branch-and-Bound nodes examined (including the root node)
LB (root)	lower bound obtained at the root node of the Branch-and-Bound tree (after adding cutting planes)
LB (final)	global lower bound at the end of the optimization
UB	best upper bound found
Gap	gap between the final upper and lower bounds :
	$gap = \frac{100(UB-LB)}{LB}$
CPU time	time spent in the Branch-and-Cut (without the Tabu Search)

Table 1. List of abbreviations

$ V $	K	$ E $	# ineq.	# B&B nodes	LB (root)	LB (final)	UB	Gap (root)	Gap (final)	CPU time (hh:mm:ss)
12	200	49	220	35	577	622	622	7.8	0.0	0:00:09
12	250	58	64	7	530	541	541	2.1	0.0	0:00:01
12	300	63	125	35	514	541	541	5.3	0.0	0:00:27
12	350	65	65	5	509	521	521	2.4	0.0	0:00:01
12	400	66	92	13	503	521	521	3.6	0.0	0:00:02
12	450	66	79	15	499	516	516	3.4	0.0	0:00:12
12	500	66	27	1	496	496	496	0.0	0.0	0:00:01
17	200	88	296	39	802	834	834	4.0	0.0	0:00:22
17	250	109	379	103	743	789	789	6.2	0.0	0:00:47
17	300	119	144	23	711	726	726	2.1	0.0	0:00:08
17	350	126	223	45	701	725	725	3.4	0.0	0:00:14
17	400	133	336	139	688	720	720	4.7	0.0	0:00:36
17	450	135	69	7	680	689	689	1.3	0.0	0:00:03
17	500	136	93	11	675	689	689	2.1	0.0	0:00:05
30	150	174	34851	44411	1057	1180	1258	19.0	6.6	10:00:00
30	200	266	25477	32175	973	1065	1065	9.5	0.0	8:26:55
30	250	328	1712	1765	894	935	935	4.6	0.0	0:31:40
30	300	372	1090	815	875	898	898	0.3	0.0	0:15:53
30	350	406	390	115	858	874	874	1.9	0.0	0:03:51
30	400	422	559	365	849	868	868	2.2	0.0	0:07:04
30	450	432	1622	1313	843	868	868	3.0	0.0	0:26:19
30	500	435	273	47	837	844	844	0.8	0.0	0:01:45
52	150	559	7877	12127	1198	1249	1339	11.8	7.2	10:00:00
52	200	821	6514	7475	1132	1165	1239	9.5	6.4	10:00:00
52	250	1040	5161	4845	1097	1125	1184	8.0	5.2	10:00:00
52	300	1149	5220	3627	1073	1096	1138	6.1	3.8	10:00:00
52	350	1230	4486	3727	1061	1082	1085	2.3	0.3	10:00:00
52	400	1287	2828	1407	1048	1066	1066	1.7	0.0	4:32:55
52	450	1313	926	195	1042	1053	1053	1.1	0.0	0:54:33
52	500	1324	1558	417	1039	1053	1053	1.3	0.0	1:49:12

Table 2. Results for real applications

K	cut	subset	ring-cut	node-partition	weighted partition	metric
150	143	250	2317	550	1151	3466
200	194	473	1074	364	481	3928
250	140	415	648	119	242	3597
300	239	561	383	117	286	3634
350	198	559	271	89	209	3160
400	170	466	115	41	101	1935
450	111	138	27	2	40	609
500	110	349	41	9	58	991

Table 3. Inequalities generated for the 52 nodes real instance

Data on the randomly generated test problems are available at the Web page <http://smg.ulb.ac.be/~bfortz/2cnbm/data.html>.

Table 2 reports results obtained for problems coming from real applications, while Table 5 reports average results obtained for randomly generated problems. The gaps are relative to the best upper bound found. The CPU times reported do not include the Tabu Search procedure, as it was run in parallel on another

$ V $	K	$ E $	# ineq.	# B&B nodes	LB (root)	LB (final)	UB	Gap (root)	Gap (final)	CPU time (hh:mm:ss)
12	200	49	190	59	565	622	622	10.1	0.0	0:00:08
12	250	58	60	11	525	541	541	3.0	0.0	0:00:01
12	300	63	77	91	513	541	541	5.5	0.0	0:00:04
12	350	65	24	9	507	521	521	2.8	0.0	0:00:00
12	400	66	54	19	502	521	521	3.8	0.0	0:00:01
12	450	66	39	51	499	516	516	3.4	0.0	0:00:02
12	500	66	11	1	496	496	496	0.0	0.0	0:00:00
17	200	88	246	81	784	834	834	6.4	0.0	0:00:20
17	250	109	299	119	738	789	789	6.9	0.0	0:00:26
17	300	119	102	23	711	726	726	2.1	0.0	0:00:04
17	350	126	147	89	701	725	725	3.4	0.0	0:00:11
17	400	133	334	279	688	719	720	4.7	0.1	0:00:38
17	450	135	69	13	678	689	689	1.6	0.0	0:00:02
17	500	136	50	17	675	689	689	2.1	0.0	0:00:02
30	150	174	18202	39697	1043	1151	1269	21.7	10.3	10:00:00
30	200	266	14858	32661	945	1035	1065	12.7	2.9	10:00:00
30	250	328	1775	2559	891	935	935	4.9	0.0	0:39:00
30	300	372	1064	1087	870	898	898	3.2	0.0	0:15:45
30	350	406	582	439	856	874	874	2.1	0.0	0:06:07
30	400	422	617	517	847	868	868	2.5	0.0	0:07:50
30	450	432	2567	2609	840	868	868	3.3	0.0	1:26:48
30	500	435	178	39	836	844	844	1.0	0.0	0:00:54
52	150	559	4674	11195	1192	1242	1339	12.3	7.8	10:00:00
52	200	821	6805	9005	1125	1158	1253	11.4	8.2	10:00:00
52	250	1040	4282	4721	1092	1119	1192	9.2	6.5	10:00:00
52	300	1149	3964	4331	1070	1094	1121	4.8	2.5	10:00:00
52	350	1230	3617	2505	1057	1078	1088	2.9	0.9	10:00:00
52	400	1287	4227	3235	1047	1062	1066	1.8	0.4	10:00:00
52	450	1313	1753	551	1042	1053	1053	1.1	0.0	1:37:25
52	500	1324	1238	377	1039	1053	1053	1.3	0.0	1:24:05

Table 4. Results for real instances without subset and ring-cut inequalities

processor. Only feasible problems appear in the tables. We also report in Table 3 the number of inequalities generated for each class of inequalities implemented, for the real instance with 52 nodes.

For 20 nodes or less, all problems could be solved to optimality. For larger problems, we remark that the problems with a small value of K are much harder, the lower bound at the root of the Branch-and-Bound tree being far from the optimum.

Problems with a large value of K are easier to solve due to the fact that these problems are closer to the two-connected network problem (without ring constraints), which can be solved efficiently using cut and node-partition inequalities (for the instances we considered). Abbreviations used in the tables are summarized in Table 1.

These instances were already considered in Fortz et al. [6]. In addition to the polyhedral analysis provided before, our new contribution consists in the addition of subset and ring-cut inequalities, as well as better separation procedures for node-partition and weighted partition inequalities. It is quite difficult

$ V $	$ E $	K	p/o	# ineq.	# B&B nodes	Gap (root)	Gap (final)	CPU time (hh:mm:ss)
10	23.0	300	2/2	94.0	10.0	6.9	0.0	0:00:02
10	28.8	350	4/4	125.0	15.0	7.7	0.0	0:00:02
10	34.6	400	5/5	197.0	28.2	10.3	0.0	0:00:05
10	40.0	450	5/5	291.2	48.6	10.2	0.0	0:00:09
10	43.2	500	5/5	231.2	45.0	10.1	0.0	0:00:07
20	65.8	200	4/4	441.0	129.0	8.1	0.0	0:01:25
20	96.5	250	4/4	1942.2	2163.0	14.0	0.0	0:14:58
20	121.8	300	5/5	10059.2	14358.2	9.7	0.0	1:31:59
20	149.6	350	5/5	1307.6	1496.2	9.3	0.0	0:11:02
20	170.2	400	5/5	1123.6	685.8	9.0	0.0	0:07:28
20	182.4	450	5/5	2868.6	2734.6	8.0	0.0	0:26:06
20	187.2	500	5/5	679.0	208.2	6.1	0.0	0:03:03
30	145.7	200	3/2	17464.7	27474.3	14.3	1.9	6:41:46
30	226.2	250	4/0	26095.0	31736.0	18.6	9.2	10:00:00
30	288.8	300	5/2	17840.8	19965.8	12.6	4.9	7:12:18
30	345.8	350	5/4	7251.2	8892.6	7.4	1.0	3:19:02
30	381.6	400	5/4	4984.0	5570.6	5.3	0.5	2:25:00
30	414.8	450	5/5	1843.2	1605.4	5.1	0.0	0:48:41
30	426.8	500	5/5	1683.0	1171.0	4.9	0.0	0:39:38
40	258.0	200	4/0	5086.0	5454.0	17.7	9.6	3:00:00
40	368.8	250	5/0	4586.8	4677.0	18.0	10.6	3:00:00
40	466.6	300	5/0	3465.0	3319.0	15.1	9.1	3:00:00
40	561.6	350	5/0	3107.4	2854.2	10.3	5.3	3:00:00
40	645.6	400	5/0	3272.0	2873.4	6.9	2.6	3:00:00
40	719.8	450	5/1	2479.4	1640.2	7.0	3.1	2:45:33
40	753.6	500	5/0	2538.2	1610.2	5.8	1.9	3:00:00
50	243.0	150	1/0	3566.0	4917.0	9.4	3.9	3:00:00
50	442.2	200	4/0	2899.8	2368.0	16.4	10.8	3:00:00
50	607.0	250	5/0	1977.0	1531.4	15.7	11.0	3:00:00
50	775.4	300	5/0	1810.6	1011.8	9.4	6.0	3:00:00
50	926.0	350	5/0	1668.0	537.8	8.3	5.3	3:00:00
50	1055.2	400	5/0	1772.8	823.0	7.3	4.7	3:00:00
50	1141.0	450	5/0	1613.2	548.6	7.6	5.1	3:00:00
50	1195.4	500	5/0	1545.8	344.2	6.5	4.4	3:00:00

Table 5. Results for random networks

to compare results presented here with those in [6], as our code was completely rewritten using ABACUS, and tested on a different hardware. However, we can measure the impact of subset and ring-cut inequalities that did not appear in [6] by disabling their separation in our new code. Results obtained for problems coming from real applications are presented in Table 4. For almost all instances, we observe that the gap at the root node and the number of nodes in the Branch-and-Bound tree are larger when ring-cut and subset inequalities are not generated. For small instances (12 and 17 nodes), the computing time decreases when ring-cut and subset inequalities are not generated, due to the expensive separation procedures. However, for difficult instances, it is worth generating these inequalities since either the total computing time or the final gap decreases.

9. Conclusion

In this paper, we study the polyhedron associated with the problem of designing at minimum cost a two-connected network such that the shortest cycle to which each edge belongs does not exceed a given length K .

Several classes of valid (and sometimes facet-defining) inequalities are proposed, and efficient algorithms to solve the corresponding separation problems are described. Numerical results obtained with a branch-and-cut algorithm integrating these separation algorithms are reported.

A. Complexity of the separation of ring-cut inequalities

In this appendix, we show that the separation problem for ring-cut inequalities is NP-complete. We first define the decision version of the problem.

Problem 1 (RING-CUT-SEP). Let $G = (V, E)$ be a graph, $K > 0$ a given constant and $x := (x_e)_{e \in E}$ a vector such that $0 \leq x_e \leq 1$ for all $e \in E$.

Does there exist a subset of nodes $W \subseteq V$, $\emptyset \neq W \neq V$, and an independent subset $S \subseteq \delta(W)$ in the ring-cut graph $RCG_{W,K}$, such that

$$x(S) + 2x(\delta(W) \setminus S) < 3 \quad ?$$

Theorem 7. *If $|V| \geq 4$, RING-CUT-SEP is NP-complete for any fixed $K \geq 3$.*

Proof. It is easy to see that RING-CUT-SEP belongs to NP. We show that the independent subset problem reduces to RING-CUT-SEP.

The independent subset problem is the following : given a graph $G = (V, E)$ and an integer p , $1 \leq p \leq |V|$, does there exist an independent subset $S \subseteq V$ of cardinality $|S| \geq p$?

The independent subset problem is known to be NP-complete (see Garey and Johnson [7]). Given an instance of this problem, we construct an instance of RING-CUT-SEP in the following way. Let $G' = (V_1 \cup V_2, E_1 \cup E_2 \cup \overline{E})$ be an undirected graph where V_1 and V_2 are two identical copies of V , E_1 and E_2 are edge sets defining complete graphs over V_1 and V_2 , and \overline{E} contains edges joining nodes in V_1 and V_2 corresponding to a same node in V . G' has $2|V|$ nodes and $|V|^2$ edges. This transformation is illustrated in Figure 6.

The length of the edges and the vector x are defined as follows :

$$d_e = \begin{cases} 1 & \text{if } e \in E_1 \cup E_2 \\ & \text{and } e \text{ does not correspond to an edge in } E, \\ \frac{1}{2} & \text{if } e \in E_1 \cup E_2 \\ & \text{and } e \text{ corresponds to an edge in } E, \\ \frac{K-1}{2} & \text{if } e \in \overline{E}. \end{cases}$$

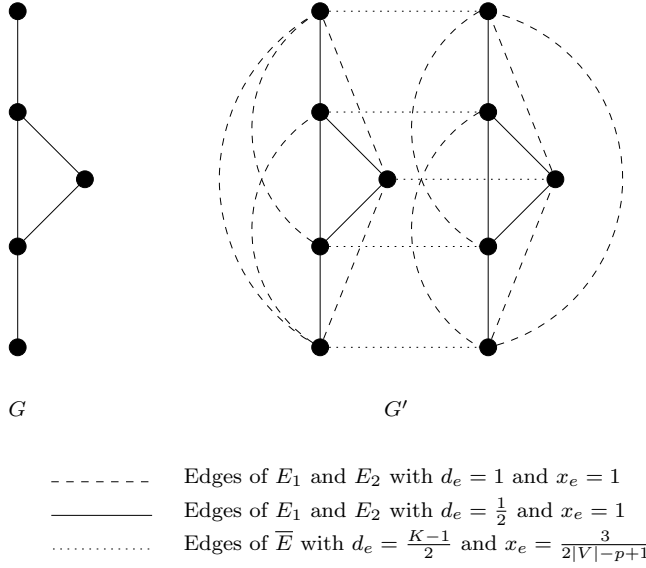


Fig. 6. Transformation of independent subset into RING-CUT-SEP

$$x_e = \begin{cases} 1 & \text{if } e \in E_1 \cup E_2, \\ \frac{3}{2|V|-p+1} & \text{if } e \in \bar{E}. \end{cases}$$

We must show that there exists an independent subset $S \subseteq V$ of cardinality $|S| \geq p$ in G if and only if there exists a subset of nodes $W \subseteq V_1 \cup V_2$, $\emptyset \neq W \neq V_1 \cup V_2$, and an independent subset $\bar{S} \subseteq \delta_{G'}(W)$ in the ring-cut graph $RCG_{W,K}$ of G' , such that $x(\bar{S}) + 2x(\delta_{G'}(W) \setminus \bar{S}) < 3$.

Consider the cut defined by $W = V_1$. Each edge in this cut corresponds to a node in V . Let $e, f \in \delta_{G'}(W)$ be two edges in the cut. If there exists a feasible cycle containing e and f , it is easy to see that $(C_{e,f})_K = E_1 \cup E_2 \cup \{e, f\}$ and that $(C_{e,f})_K$ is two-connected, since E_1 and E_2 define complete subgraphs, since each edge in E_1 and E_2 has a length less than or equal to one, and since $K \geq 3$. A feasible cycle containing e and f must use at least one edge in E_1 and one edge in E_2 . The only possibility is thus to use one edge of length $\frac{1}{2}$ in E_1 and in E_2 , leading to a cycle of length equal to K . These edges may be chosen provided there exists an edge between the nodes corresponding to e and f in E . We can conclude that $(C_{e,f})_K$ is two-connected if and only if there is an edge in E between the nodes corresponding to e and f . Therefore, the ring-cut graph $RCG_{W,K}$ is isomorphic to G .

Suppose $S \subseteq V$ is an independent subset of cardinality $|S| \geq p$ in G , and let again $W = V_1$. Since $RCG_{W,K}$ is isomorphic to G , there exists $\bar{S} \subseteq \delta_{G'}(W)$ such that $|\bar{S}| = |S| \geq p$ and \bar{S} is an independent subset in the ring-cut graph $RCG_{W,K}$. By the definition of x , we obtain that

$$x(\bar{S}) + 2x(\delta_{G'}(W) \setminus \bar{S}) = 2x(\delta_{G'}(W)) - x(\bar{S})$$

$$\begin{aligned}
&= \frac{3(2|V| - |\bar{S}|)}{2|V| - p + 1} \\
&\leq \frac{3(2|V| - p)}{2|V| - p + 1} \\
&< 3,
\end{aligned}$$

and W and \bar{S} define the requested violated ring-cut inequality.

If there exists a subset of nodes $W \subseteq V_1 \cup V_2$, $\emptyset \neq W \neq V_1 \cup V_2$, and an independent subset $\bar{S} \subseteq \delta_{G'}(W)$ in the ring-cut graph $RCG_{W,K}$ of G' , such that $x(\bar{S}) + 2x(\delta_{G'}(W) \setminus \bar{S}) < 3$, then we must have $x(\delta_{G'}(W)) < 3$, since $\bar{S} \subseteq \delta_{G'}(W)$. The edge sets E_1 and E_2 define complete subgraphs with edges of value 1, and therefore each cut separating two nodes of V_1 or two nodes of V_2 has a value greater than or equal to $|V| - 1$. The only possibility for W is then to have $\delta_{G'}(W) = \delta_{G'}(V_1)$, and the ring-cut graph $RCG_{W,K}$ is again isomorphic to G . Thus there exists an independent subset $S \subseteq V$ in G such that $|S| = |\bar{S}|$. Moreover,

$$3 > x(\bar{S}) + 2x(\delta_{G'}(W) \setminus \bar{S}) = \frac{3(2|V| - |\bar{S}|)}{2|V| - p + 1}.$$

It follows that $|S| = |\bar{S}| > p - 1$, thus $|S| \geq p$ and S is the requested independent subset.

References

1. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley, 1974.
2. F. Barahona. Separating from the dominant of the spanning tree polytope. *Op. Research Letters*, 12:201–203, 1992.
3. W.H. Cunningham. Optimal attack and reinforcement of a network. *Journal of ACM*, 32:549–561, 1985.
4. M. Didi Biha and A.R. Mahjoub. k -edge connected polyhedra on series-parallel graphs. *Operations Research Letters*, 19:71–78, 1996.
5. B. Fortz. *Design of Survivable Networks with Bounded Rings*, volume 2 of *Network Theory and Applications*. Kluwer Academic Publishers, 2000.
6. B. Fortz, M. Labbé, and F. Maffioli. Solving the two-connected network with bounded meshes problem. *Operations Research*, 48(6):866–877, 2000.
7. M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.
8. R.E. Gomory and T.C. Hu. Multi-terminal network flows. *SIAM J. Appl. Math.*, 9:551–570, 1961.
9. M. Grötschel and C.L. Monma. Integer polyhedra arising from certain design problems with connectivity constraints. *SIAM J. Discrete Math.*, 3:502–523, 1990.
10. M. Grötschel, C.L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40(2):309–330, 1992.
11. M. Grötschel, C.L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM J. Optimization*, 2(3):474–504, 1992.
12. M. Grötschel, C.L. Monma, and M. Stoer. *Design of Survivable Networks*, volume 7 on Network models of *Handbooks in OR/MS*, chapter 10, pages 617–672. North-Holland, 1995.

13. A.R. Mahjoub. Two-edge connected spanning subgraphs and polyhedra. *Mathematical Programming*, 64:199–208, 1994.
14. A. Sassano. On the facial structure of the set covering polytope. *Mathematical Programming*, 44:181–202, 1989.
15. M. Stoer. *Design of Survivable Networks*, volume 1531 of *Lecture Notes in Mathematics*. Springer-Verlag, 1992.
16. S. Thienel. *ABACUS—A Branch-And-Cut System*. PhD thesis, Universität zu Köln, 1995.