A Branch-and-Cut Algorithm for the k-Edge Connected Subgraph Problem

F. Bendali, I. Diarrassouba, and A. R. Mahjoub
Laboratoire LIMOS, CNRS UMR 6158, Université Blaise Pascal–Clermont II, Complexe Scientifique des Cézeaux, 63177 Aubière Cedex, France

M. Didi Biha
Laboratoire d'Analyse Non Linéaire et Géométrie, Université d'Avignon, 339, Chemin des Meinajariès, 84911 Avignon Cedex 9, France

J. Mailfert
Laboratoire LIMOS, CNRS UMR 6158, Université d'Auvergne, 49 Bd François Mitterand, 63000 Clermont-Ferrand, France

In this article, we consider the k-edge connected subgraph problem from a polyhedral point of view. We introduce further classes of valid inequalities for the associated polytope and describe sufficient conditions for these inequalities to be facet defining. We also devise separation routines for these inequalities and discuss some reduction operations that can be used in a preprocessing phase for the separation. Using these results, we develop a Branch-and-Cut algorithm and present some computational results. **NETWORKS, Vol. 55(1), 13–32 2010**

Keywords: k-edge connected graph; polytope; facet; separation; reduction operations; branch-and-cut

1. INTRODUCTION

One of the main concerns when designing telecommunication networks is to compute network topologies that provide a sufficient degree of survivability. Survivable networks must satisfy some connectivity requirements that is, networks that are still functional after the failure of certain links. As pointed out in [31] (see also [29]), the topology that seems to be very efficient (and needed in practice) is the uniform topology, that is to say that corresponding to networks that survive after the failure of $k-1$ or fewer edges, for some $k \geq 2$. The two-connected topology ($k = 2$) provides an adequate level of survivability since most failure usually can be repaired relatively quickly. However, for many applications, it may be necessary to provide a higher level of connectivity. In this article, we consider this variant of the survivable network design problem.

A graph $G = (V, E)$ is called k-edge connected (where k is a positive integer) if for every pair of nodes $i, j \in V$, there are at least k edge-disjoint paths between i and j. Given a graph $G = (V, E)$ and a weight function w on E that associates with an edge $e \in E$ the weight $w(e) \in \mathbb{R}$, the *k-edge connected subgraph problem* (kECSP for short) is to find a k-edge connected spanning subgraph $H = (V, F)$ of G such that $\sum_{e \in F} w(e)$ is minimum.

The kECSP is NP-hard for $k \geq 2$ ([21]). When $k = 1$, the kECSP is nothing but the minimum spanning tree problem and can be solved in polynomial time. The kECSP has been extensively studied when $k = 2$ [4, 18, 20, 29–34]. It has, however, received little attention when $k \geq 3$.

In this article, we consider the k-edge connected subgraph problem from a polyhedral point of view. We introduce further classes of valid inequalities for the associated polytope and describe sufficient conditions for these inequalities to be facet defining. We also devise separation heuristics for these inequalities and discuss some reduction operations that can be used in a preprocessing phase for the separation. Using these results, we develop a Branch-and-Cut algorithm and present some computational results.

We will denote a graph by $G = (V, E)$ where V is the *node set* and E is the *edge set*. Given a graph $G = (V, E)$ and an edge subset $F \subseteq E$, the 0-1 vector $x^F \in \mathbb{R}^E$ such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ if $e \in E \setminus F$ is called the *incidence vector* of F. Let kECSP(G) be the convex hull of

Received November 2006; accepted June 2008
Correspondence to: A. R. Mahjoub; e-mail: mahjoub@lamsade.dauphine.fr; Laboratoire LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775, Paris 16, France
DOI 10.1002/net.20310
Published online 6 March 2009 in Wiley InterScience (www.interscience.wiley.com).

the incidence vectors of the k-edge connected subgraphs of G, i.e.,

$$k\text{ECSP}(G) = \text{conv}\{x^F \in \mathbb{R}^E \mid F \subseteq E \text{ and } (V,F) \text{ is } k\text{-edge connected subgraph of } G\}.$$

The kECSP(G) is called the *k-edge connected subgraph polytope* of G.

Let $G = (V,E)$ be a graph. Given $w : E \rightarrow \mathbb{R}$ and F a subset of E, $w(F)$ will denote $\sum_{e\in F} w(e)$. For $W \subseteq V$, we let $\bar{W} = V \setminus W$. If $W \subset V$ is a node subset of G, then the set of edges that have only one node in W is called a *cut* and denoted by $\delta(W)$. We will write $\delta(v)$ for $\delta(\{v\})$. If x^F is the incidence vector of the edge set F of a k-edge connected spanning subgraph of G, then x^F satisfies the following inequalities:

$$x(e) \geq 0 \qquad \text{for all } e \in E, \tag{1.1}$$

$$x(e) \leq 1 \qquad \text{for all } e \in E, \tag{1.2}$$

$$x(\delta(W)) \geq k \qquad \text{for all } W \subset V, W \neq V, W \neq \emptyset. \tag{1.3}$$

Conversely, any integer solution of the system defined by inequalities (1.1)–(1.3) is the incidence vector of the edge set of a k-edge connected subgraph of G. Constraints (1.1) and (1.2) are called *trivial inequalities* and constraints (1.3) are called *cut inequalities*. We will denote by $P(G,k)$ the polytope given by inequalities (1.1)–(1.3).

The kECSP has been studied by Grötschel and Monma [24] and Grötschel et al. [25, 26] within the framework of a more general survivability model. In particular, Grötschel and Monma [24] studied the dimension of the polytope associated with that model and some basic facets. It follows from their results that kECSP is full dimensional if G is $(k+1)$-edge connected. In [25], Grötschel et al. studied further polyhedral aspects of that model. They also devised cutting plane algorithms and presented some computational results. A complete survey of that model and related network design problems can be found in [29].

In [7], Chopra studies the k-edge connected subgraph problem for k odd when multiple copies of an edge may be used. In particular, he characterizes the associated polyhedron for outerplanar graphs (a graph is outerplanar if it can be drawn in the plane as a cycle with noncrossing chords). This polyhedron has been previously studied by Cornuéjols et al. [8]. They showed that if a graph is series-parallel [a graph is series-parallel if it can be obtained from a single edge by iterative application of the two operations: (i) addition of a parallel edge and (ii) subdivision of an edge] and $k = 2$, then the polyhedron is completely described by the nonnegativity and cut inequalities. In [14], Didi Biha and Mahjoub give a complete description of kECSP(G) for all k when G is series-parallel. In particular, they show that if G is series-parallel and k is even, then kECSP(G) $= P(G,k)$. Didi Biha and Mahjoub study in [16] the extreme points of $P(G,k)$. They introduce an ordering on the fractional extreme points of $P(G,k)$ and describe some structural properties of the minimal extreme points with respect to that ordering. Using these results, they give sufficient conditions for $P(G,k)$ to be integral.

Much work has been done on 2ECSP(G). In [3], Baïou and Mahjoub study the Steiner two-edge connected subgraph polytope. This has been generalized by Didi Biha and Mahjoub [15] to the Steiner k-edge connected subgraph polytope for k even. Mahjoub [32] introduces a general class of valid inequalities for 2ECSP(G). Boyd and Hao [6] describe a class of "comb inequalities" which are valid for 2ECSP(G). This class and that introduced by Mahjoub [32] are special cases of a more general class of inequalities given by Grötschel et al. [25] for the general survivable network polytope. In [4], Barahona and Mahjoub characterize the polytope 2ECSP(G) for the class of Halin graphs. Kerivin et al. [30] describe a general class of valid inequalities for 2ECSP(G) that generalizes the so-called F-partition inequalities [32]. They also develop a Branch-and-Cut algorithm for the 2ECSP, based on these inequalities and the trivial and cut inequalities. In [5], Bienstock et al. describe structural properties of the optimal solution of kECSP when the weight function satisfies the triangle inequalities (i.e., $w(e_1) \leq w(e_2) + w(e_3)$ for every three edges e_1, e_2, e_3 defining a triangle). In particular, they show that every node of a minimum weight k-edge connected subgraph has degree k or $k+1$. This generalizes results given by Monma et al. [34] for the case when $k = 2$. In [9, 10], Coullard et al. study the Steiner two-node connected subgraph problem. They devise in [9] a linear time algorithm for this problem on some special classes of graphs. Moreover in [10], they characterize the dominant of the polytope associated with this problem on the graphs which do not have K_4 (the wheel on four nodes) as a minor. In [19], Fonlupt and Naddef characterize the class of graphs for which the system given by inequalities (1.1) and (1.3), when $k = 2$, defines the convex hull of the incidence vectors of the tours of G (a tour is a cycle going at least once through each node. Here a cycle can use the same node or the same edge more than once).

The article is organized as follows. In the following section, we introduce some classes of valid inequalities and describe sufficient conditions for these inequalities to be facet defining. In Section 3, we discuss some graph reduction operations. In Section 4, we describe separation routines for the inequalities described in Section 2 and develop a Branch-and-Cut algorithm for the kECSP. Our computational results are presented in Section 5, and finally some concluding remarks are given in Section 6. Some of the proofs of section 2 are given in an appendix.

In the rest of this section we give more definitions and notations. The graphs we consider are finite, undirected, loopless, and connected. Given a graph $G = (V,E)$, if $e \in E$ is an edge with endnodes u and v, we also write uv to denote e. Given a node subset W, the cut $\delta(W)$ is said to be *proper* if $|W| \geq 2$ and $|V \setminus W| \geq 2$. If W and W' are two disjoint subsets of V, $[W, W']$ will denote the set of edges of G having one endnode in W and the other one in W'. If $\pi = (V_1, \ldots, V_p)$, $p \geq 2$, is a partition of V, then we denote by $\delta(\pi)$ the set of edges having their endnodes in different sets. We may also write $\delta(V_1, \ldots, V_p)$ for $\delta(\pi)$. Note that for $W \subset V$, $\delta(W) = \delta(W, \bar{W})$. In the notation, we will specifiy the graph as a subscript (that is, we will write $\delta_G(W)$, $\delta_G(\pi)$, $\delta_G(V_1, \ldots, V_p)$) whenever the considered graphs may not be clearly deduced from the context.

For all $F \subseteq E$, $V(F)$ will denote the set of nodes incident to the edges of F. For $W \subset V$, we denote by $E(W)$ the set of edges of G having both endnodes in W and $G[W]$ the subgraph induced by W. Given an edge $e = uv \in E$, *contracting* e consists in deleting e, identifying the nodes u and v and in preserving all adjacencies. Contracting a node subset W consists in identifying all the nodes of W and preserving the adjacencies. Given a partition $\pi = (V_1, \ldots, V_p)$, $p \geq 2$, we will denote by G_π the subgraph induced by π, that is, the graph obtained from G by contracting the sets V_i, for $i = 1, \ldots, p$. Note that the edge set of G_π is the set $\delta(V_1, \ldots, V_p)$. Given a solution $\bar{x} \in P(G, k)$, an inequality $ax \geq \alpha$ is said to be *tight* for $\bar{x}$ if $a\bar{x} = \alpha$.

2. FACETS OF *k*ECSP(*G*)

In this section, we present three classes of valid inequalities for kECSP(G). We describe some conditions for these inequalities to be facet defining. But first, we give the following lemmas, which will be frequently used in this section.

Lemma 2.1. *If an inequality $ax \geq \alpha$ is different from the trivial inequalities and defines a facet of kECSP(G), then $a(e) \geq 0$ for all $e \in E$ and $\alpha > 0$.*

Proof. Suppose that $a(e) < 0$ for some edge $e \in E$. As $ax \geq \alpha$ is different from the trivial inequality $x(e) \leq 1$, there must exist a solution $F \subseteq E$ of the kECSP which does not contain e and such that $ax^F = \alpha$. Let $F' = F \cup \{e\}$. Obviously, F' also induces a solution of the kECSP. However, as $a(e) < 0$, we have that $ax^{F'} = ax^F + a(e) < \alpha$, contradiction.

In consequence, $a(e) \geq 0$ for all $e \in E$. Moreover, as $ax \geq \alpha$ is facet defining, one should have $a(f) > 0$ for at least one edge f of E. As $ax \geq \alpha$ is different from $x(f) \geq 0$, there exists a solution $\tilde{F}$ of the kECSP which contains f and such that $ax^{\tilde{F}} = \alpha$. This yields $\alpha > 0$. ▪

Lemma 2.2. *Let $G = (V, E)$ be a k-edge connected graph and $e_0 = u_0v_0$ be an edge of G such that every cut $\delta(U)$ of G containing e_0, except eventually $\delta(u_0)$, is such that $|\delta(U)| \geq k + 1$. If G' is a graph obtained from G by deleting e_0 and adding an edge f incident to u_0, then G' is k-edge connected.*

Proof. Let $\delta_{G'}(U')$ be a cut of G'. If $\delta_{G'}(U')$ does not separate u_0 and v_0, then, as G is k-edge connected, we have that $|\delta_{G'}(U')| \geq k$. If this is not the case and $U' \neq \{u_0\}$, then $\delta_G(U')$ contains at least $k+1$ edges and hence $|\delta_{G'}(U')| \geq k$. Finally, if $U' = \{u_0\}$, as G is k-edge connected and $\delta_{G'}(u_0) = (\delta_G(u_0) \setminus \{e_0\}) \cup \{f\}$, we have that $|\delta_{G'}(u_0)| \geq k$. ▪

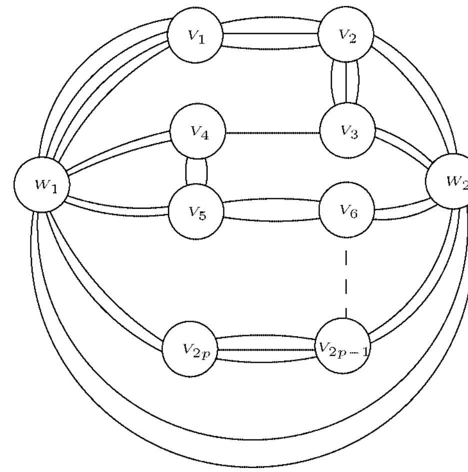

FIG. 1. An odd path configuration with $k = 3$ and p even.

2.1. Odd Path Inequalities

Let $G = (V, E)$ be a $(k + 1)$-edge connected graph and $\pi = (W_1, W_2, V_1, \ldots, V_{2p})$ a partition of V with $p \geq 2$. Let $I_1 = \{4r, 4r + 1, r = 1, \ldots, \lceil \frac{p}{2} \rceil - 1\}$ and $I_2 = \{2, \ldots, 2p - 1\} \setminus I_1$. We say that π induces an *odd path configuration* if

1. $|[V_i, W_j]| = k - 1$ for $(i, j) \in (I_1 \times \{1\}) \cup (I_2 \times \{2\})$,
2. $|[W_1, W_2]| \leq k - 1$,
3. $\delta(V_i) = [V_i, W_1] \cup [V_{i-1}, V_i] \cup [V_i, V_{i+1}]$ (respectively $\delta(V_i) = [V_i, W_2] \cup [V_{i-1}, V_i] \cup [V_i, V_{i+1}]$) if $i \in I_1$ (respectively $i \in I_2$),
4. $\delta(V_1) = [W_1, V_1] \cup [V_1, V_2]$ and $\delta(V_{2p}) = [W_1, V_{2p}] \cup [V_{2p-1}, V_{2p}]$ (respectively $\delta(V_{2p}) = [W_2, V_{2p}] \cup [V_{2p-1}, V_{2p}]$) if p is even (respectively odd) (see Fig. 1 for $k = 3$ and p even).

Note that by conditions 3 and 4, we have that $[V_l, V_t] = \emptyset$ for all $l, t \in \{1, \ldots, 2p\}$ and $|l - t| > 1$. Let $C = \bigcup_{i=1}^{2p-1} [V_i, V_{i+1}]$. Thus C can be seen as an odd path of extremities V_1 and V_{2p} in the graph G_π. With an odd path configuration we associate the inequality

$$x(C) \geq p. \tag{2.1}$$

Inequalities of type (2.1) will be called *odd path inequalities*. We have the following.

Theorem 2.1. *Inequality* (2.1) *is valid for kECSP(G).*

Proof. As $|[V_i, W_j]| = k - 1$ and $x(\delta(V_i)) \geq k$ is valid for kECSP(G), for $(i, j) \in (I_1 \times \{1\}) \cup (I_2 \times \{2\})$, we have

$$x([V_{2s-1}, V_{2s}]) + x([V_{2s}, V_{2s+1}]) \geq 1 \text{ for } s = 1, \ldots, p - 1, \tag{2.2}$$

$$x([V_{2s}, V_{2s+1}]) + x([V_{2s+1}, V_{2s+2}]) \geq 1 \text{ for } s = 1, \ldots, p - 1. \tag{2.3}$$

By multiplying each inequality (2.2) (respectively inequality (2.3)) corresponding to $s \in \{1, \ldots, p - 1\}$ by $\frac{p-s}{p}$ (respectively $\frac{s}{p}$) and summing these inequalities, we obtain

$$\sum_{i \in I} x([V_i, V_{i+1}]) + \sum_{i \in \bar{I}} \frac{p-1}{p} x([V_i, V_{i+1}]) \geq p - 1, \tag{2.4}$$

where $I = \{2, 4, 6, \ldots, 2p-2\}$ and $\bar{I} = \{1, \ldots, 2p-1\} \setminus I$.

By considering the cut inequality induced by $W_1 \cup V_1 \cup (\bigcup_{i\in I_1} V_i)$ (respectively $W_1 \cup V_1 \cup (\bigcup_{i\in I_1} V_i) \cup V_{2p}$) if p is odd (respectively even) we have

$$x([W_1, W_2]) + \sum_{i\in\bar{I}} x([V_i, V_{i+1}]) \geq k.$$

As $|[W_1, W_2]| \leq k-1$, it follows that

$$\frac{1}{p}\sum_{i\in\bar{I}} x([V_i, V_{i+1}]) \geq \frac{1}{p}. \tag{2.5}$$

By summing inequalities (2.4) and (2.5) and rounding up the right hand side, we get inequality (2.1). ∎

In what follows, we describe necessary conditions for inequality (2.1) to be facet defining. For this, we first give a technical lemma.

Lemma 2.3. *Let $\pi = (W_1, W_2, V_1, \ldots, V_{2p})$, $p \geq 2$, be a partition of V which induces an odd path configuration and F a solution of the kECSP. Let $V_r, \ldots, V_s$, with $2 \leq r < s \leq 2p-1$, be a sequence of node sets of π. Then F must contain at least $\lceil \frac{s-r+1}{2} \rceil$ edges from C.*

Proof. As $|[W_1, V_i]| = k-1$ for all $i \in \{r, \ldots, s\} \cap I_1$ and $|[W_2, V_i]| = k-1$ for all $i \in \{r, \ldots, s\} \cap I_2$, F must contain at least one edge from each set $\delta(V_i) \cap C$, $i \in \{r, \ldots, s\}$. Thus the statement follows. ∎

Theorem 2.2. *Inequality (2.1) defines a facet for kECSP(G) only if*

(a) *$[V_1, W_1] \neq \emptyset$ and $[V_{2p}, W_1] \neq \emptyset$ (respectively $[V_{2p}, W_2] \neq \emptyset$) if p is even (respectively odd),*
(b) *$[V_i, V_{i+1}] \neq \emptyset$ for $i = 1, \ldots, 2p-1$.*

Proof.

(a) Suppose for instance that p is even and $[V_1, W_1] = \emptyset$ (the proof is similar if either $[V_{2p}, W_1] = \emptyset$ or p is odd and $[V_{2p}, W_2] = \emptyset$). By contracting the sets V_1, V_2, W_2, we obtain a smaller odd path configuration with $2p$ elements. Let

$$x(C') \geq p - 1 \tag{2.6}$$

be the corresponding odd path inequality. As $\delta(V_2) = [V_1, V_2] \cup [V_2, V_3] \cup [V_2, W_2]$ and $|[V_2, W_2]| = k-1$, by the cut constraint on V_2, we have that

$$x([V_1, V_2]) + x([V_2, V_3]) \geq 1 \tag{2.7}$$

is valid for kECSP(G). By adding (2.6) and (2.7), we get $x(C) \geq p$, which implies that (2.1) cannot be facet defining.

(b) Suppose that $[V_i, V_{i+1}] = \emptyset$ for some $i \in \{1, \ldots, 2p-1\}$. We will show in the followining that any solution F of the kECSP whose the incidence vector x^F satisfies (2.1) with equality, intersects $[V_{i-1}, V_i]$ in exactly one edge. To this end, we will distinguish two cases.

CASE 1. $i, i+1 \in I_1$ (the proof is similar if $i, i+1 \in I_2$). By Lemma 2.3 the edge set $F' = F \cap C$ must cover the node sets $V_2, \ldots, V_{i-2}$ by at least $\lceil \frac{i-3}{2} \rceil$ edges and the sets $V_{i+1}, \ldots V_{2p-1}$ by at least $\lceil \frac{2p-i-1}{2} \rceil$ edges. As $i, i+1 \in I_1$, and then i is even, F' must use, in consequence, at least $(\frac{i}{2} - 1) + (p - \frac{i}{2}) = p - 1$ edges from $C \setminus [V_{i-1}, V_i]$. As $\delta(V_i) = [V_{i-1}, V_i] \cup [V_i, W_1]$ and $|[V_i, W_1]| = k-1$, F contains at least one edge from $[V_{i-1}, V_i]$. As x^F satisfies (2.1) with equality, it follows that F contains exactly one edge from $[V_{i-1}, V_i]$.

CASE 2. $i \in I_1$ and $i+1 \in I_2$ (the proof is similar if $i \in I_2$ and $i+1 \in I_1$). First note that in this case i is odd. By Lemma 2.3, F must cover the node sets $V_2, \ldots, V_{i-2}$ by at least $\lceil \frac{i-3}{2} \rceil = \frac{i-3}{2}$ edges from C and the node sets $V_{i+1}, \ldots V_{2p-1}$ by at least $\lceil \frac{2p-i-1}{2} \rceil = \frac{2p-i-1}{2}$ edges from C. Hence F uses at least $\frac{i-3}{2} + \frac{2p-i-1}{2} = p-2$ edges from C. Moreover, observe that if exactly $p-2$ edges of C are used by F, then these edges should be between consecutive node sets of the form $[V_{2s}, V_{2s+1}]$, with $s \in \{1, \ldots, p-1\} \setminus \{\frac{i-1}{2}\}$. However, in this case, to satisfy the cut inequality induced by the node set $W_1 \cup (\bigcup_{r\in I_1} V_r) \cup V_{2p}$ (respectively $W_1 \cup (\bigcup_{r\in I_1} V_r)$) if p is even (respectively odd), F must contain at least one more edge from $C \setminus [V_{i-1}, V_i]$ between two consecutive sets of the form $[V_{2s-1}, V_{2s}]$, with $s \in \{1, \ldots, p-1\} \setminus \{\frac{i-1}{2}\}$. In consequence, F contains at least $p-1$ edges from $C \setminus [V_{i-1}, V_i]$. As $|F \cap [V_{i-1}, V_i]| \geq 1$ and x^F satisfies (2.1) with equality, we then have that $|F \cap [V_{i-1}, V_i]| = 1$.

In consequence, for any solution $F \subseteq E$ of the kECSP, if x^F satisfies (2.1) with equality, it also satisfies the equation $x(\delta(V_i)) = k$. As kECSP(G) is full dimensionnal and (2.1) is not a positive multiple of $x(\delta(V_i)) \geq k$, (2.1) cannot define a facet. ∎

Now we give sufficient conditions for inequality (2.1) to be facet defining. For this, let us denote by Γ the set of edges of G which are not in C, that is, $\Gamma = E \setminus C$. Moreover, if $[V_i, V_{i+1}] \neq \emptyset$, we let e_i denote a fixed edge of $[V_i, V_{i+1}]$, for $i = 1, \ldots, 2p-1$.

Theorem 2.3. *Inequality (2.1) defines a facet for kECSP(G) if the following hold.*

1. *Condition (b) of Theorem 2.2 holds,*
2. *The subgraphs $G[W_1]$, $G[W_2]$ and $G[V_i]$, for $i = 1, \ldots, 2p$, are $(k+1)$-edge connected,*
3. *$|[W_1, W_2]| = k-1$, $|[V_1, W_1]| = k$ and $|[V_{2p}, W_1]| = k$ (respectively $|[V_{2p}, W_2]| = k$) if p is even (respectively odd).*

Proof. We will show the result for p even (the proof is similar if p is odd).

Let $E_0 = \bigcup_{s=1}^{p}[V_{2s-1}, V_{2s}]$, $E_1 = \bigcup_{s=1}^{p-1}[V_{2s}, V_{2s+1}]$, $\bar{E} = \delta(\pi) \setminus (E_0 \cup E_1)$, $\tilde{E} = E \setminus (E_0 \cup E_1 \cup \bar{E})$. Inequality (2.1) can then be written as

$$x(E_0) + x(E_1) \geq p. \quad (2.8)$$

Suppose that conditions 1–3 above hold. We first give a claim that will be useful in the proof.

Claim. *If D is a subset of edges which covers the node sets $V_2, \ldots, V_{2p-1}$, contains at least one edge of $[V_{i_0}, V_{i_0+1}]$ for some $i_0 \in \{1, 3, \ldots, 2p-1\}$ and such that $D \cap \Gamma = \emptyset$, then $D \cup \Gamma$ induces a k-edge connected subgraph of G.*

Proof. Let $F = D \cup \Gamma$. Let $\bar{G}$ be the graph induced by F and $\bar{G}'$ the graph obtained from $\bar{G}$ by contracting the node sets $W_1, W_2, V_1, \ldots, V_{2p}$. Let $w_1, w_2, v_1, \ldots, v_{2p}$ be the nodes of $\bar{G}'$ where w_j (respectively v_i) corresponding to W_j (respectively V_i) for $j = 1, 2$ (respectively $i = 1, \ldots, 2p$). As by condition 2, the subgraphs of $\bar{G}$ induced by $W_1, W_2, V_1, \ldots, V_{2p}$ are $(k+1)$-edge connected, to show the claim, it suffices to show that $\bar{G}'$ is k-edge connected. Let $\delta_{\bar{G}'}(W)$ be a cut of $\bar{G}'$.

If, say, $w_1 \in W$ and $w_2 \in \bar{W}$, then $[w_1, w_2] \subseteq \delta_{\bar{G}'}(W)$. If $\delta_{\bar{G}'}(W)$ separates v_{i_0} and v_{i_0+1}, as D intersects $[V_{i_0}, V_{i_0+1}]$, and by condition 3), $|[W_1, W_2]| = k-1$, we have that $|\delta_{\bar{G}'}(W)| \geq k$. If $v_{i_0}, v_{i_0+1} \in W$, then $[\{v_{i_0}, v_{i_0+1}\}, w_2] \subseteq \delta_{\bar{G}'}(W)$. Since $|[\{v_{i_0}, v_{i_0+1}\}, w_2]| \geq k-1 \geq 1$, this yields $|\delta_{\bar{G}'}(W)| \geq k$.

Now if $w_1, w_2 \in W$ (or $w_1, w_2 \in \bar{W}$), then $\delta_{\bar{G}'}(W)$ contains at least two edge sets of the form $[v_i, w_j]$ with $(i,j) \in (I_1 \times \{1\}) \cup (I_2 \times \{2\})$. As $|[v_i, w_j]| = k-1$, we have that $|\delta_{\bar{G}'}(W)| \geq k$. ∎

Let us denote inequality (2.8) by $ax \geq \alpha$ and $\mathcal{F} = \{x \in k\text{ECSP}(G) \mid ax = \alpha\}$. Let $S = \Gamma \cup \{e_{2s-1}, s = 1, \ldots, p\}$. From the claim above, we can see that S induces a k-edge connected subgraph of G. Moreover, x^S satisfies (2.8) with equality, which implies that $\mathcal{F}$ is a proper face of kECSP(G). Now suppose that there exists a non trivial facet defining inequality $bx \geq \beta$ such that $\mathcal{F} \subseteq \{x \in k\text{ECSP}(G) | bx = \beta\}$. By Lemma 2.1, we have that $\beta > 0$, and hence we may suppose that $\beta = \alpha$. As G is $(k+1)$-edge connected and thus kECSP(G) is full dimensional, it suffices to show that $b = a$.

Let $e \in [V_{2s-1}, V_{2s}] \setminus \{e_{2s-1}\}$ for some $s \in \{1, \ldots, p\}$ and $S_1 = (S \setminus \{e_{2s-1}\}) \cup \{e\}$. By the claim above, S_1 induces a k-edge connected subgraph of G. Moreover, $ax^{S_1} = \alpha$. It then follows that $bx^{S_1} = \alpha$, implying that

$$b(e) = \rho_{2s-1} \text{ for all } e \in [V_{2s-1}, V_{2s}], \text{ for } s = 1, \ldots, p,$$
$$\text{for some } \rho_{2s-1} \in \mathbb{R}, \rho_{2s-1} \neq 0. \quad (2.9)$$

Similarly, for an edge $e \in [V_{2s}, V_{2s+1}] \setminus \{e_{2s}\}$ for some $s \in \{1, \ldots, p-1\}$ one can consider the edge sets $S_2 = \Gamma \cup (\bigcup_{i=1}^{p-1}\{e_{2i}\}) \cup \{e_1\}$ and $S_3 = (S_2 \setminus \{e_{2s}\}) \cup \{e\}$. We can see by the claim above that S_2 and S_3 induce k-edge connected subgraphs of G. As, $ax^{S_2} = ax^{S_3} = \alpha$, it follows that $bx^{S_2} = bx^{S_3} = \alpha$ and then

$$b(e) = \rho_{2s} \text{ for all } e \in [V_{2s}, V_{2s+1}], \text{ for } s = 1, \ldots, p-1,$$
$$\text{for some } \rho_{2s} \in \mathbb{R}, \rho_{2s} \neq 0. \quad (2.10)$$

Consider the edge sets $S_4 = (S_2 \setminus \{e_1\}) \cup \{e_{2s-1}\}$ and $S_5 = (S_2 \setminus \{e_1, e_{2s}\}) \cup \{e_{2s-1}, e_{2s+1}\}$ for some $s \in \{1, \ldots, p-1\}$. By the claim above, S_4 and S_5 induce k-edge connected subgraphs of G. Since $ax^{S_4} = ax^{S_5} = \alpha$, $bx^{S_4} = bx^{S_5} = \alpha$ and hence

$$b(e_1) = b(e_{2s}) = b(e_{2s+1}), \text{ for } s = 1, \ldots, p-1. \quad (2.11)$$

From (2.9), (2.10), and (2.11), it follows that $b(e)$ is the same for every edge $e \in E_0 \cup E_1$. As $ax^S = bx^S = \alpha$, we get $b(e) = 1$ for all $e \in E_0 \cup E_1$.

Now we are going to show that $b(e) = 0$ for all $e \in \tilde{E} \cup \bar{E}$. For this, first consider an edge $f \in \tilde{E}$. From condition 2, $S_f = S \setminus \{f\}$ induces a k-edge connected subgraph of G. Moreover, x^{S_f} satisfies (2.8) with equality. Hence $ax^{S_f} = \alpha = bx^{S_f}$. This implies that $b(f) = bx^S - bx^{S_f} = 0$. Now let $e \in [V_i, W_j]$ for $(i,j) \in (I_1 \cup \{1, 2p\} \times \{1\}) \cup (I_2 \times \{2\})$ and $S_6 = (S_2 \setminus \{e_1\}) \cup \{e_{i-1}\}$ (respectively $S_6 = (S_2 \setminus \{e_1\}) \cup \{e_i\}$) if i is even (respectively odd). From the claim above, we have that S_6 and $S'_6 = S_6 \setminus \{e\}$ induce k-edge connected subgraphs of G and that their incidence vectors satisfy $ax \geq \alpha$ with equality. Hence $b(e) = bx^{S_6} - bx^{S'_6} = 0$.

For all $e \in [W_1, W_2]$, by the claim above, the edge set $S_7 = S \setminus \{e\}$ induces a k-edge connected subgraph of G. Moreover, x^{S_7} satisfies $ax \geq \alpha$ with equality. Hence $ax^{S_7} = \alpha$ and $bx^{S_7} = bx^S = \alpha$. Thus we obtain $b(e) = 0$ for all $e \in [W_1, W_2]$.

Consequently, $b(e) = 0$ for all $e \in E \setminus C$, which terminates the proof of the theorem. ∎

2.2. *Lifting Procedure for Odd Path Inequalities*

In what follows we are going to describe a lifting procedure for the odd path inequalities. This will permit to extend these inequalities to a more general class of valid inequalities. But first we give the following lemma which easily follows from the general lifting procedure presented in [35].

Lemma 2.4. *Let $G = (V, E)$ be a graph and $ax \geq \alpha$ a valid inequality for kECSP(G). Let $G' = (V, E')$ be a graph obtained from G by adding an edge e, that is $E' = E \cup \{e\}$. Then the inequality*

$$ax + a(e)x(e) \geq \alpha \quad (2.12)$$

is valid for kECSP(G') where $a(e) = \alpha - \gamma$ with $\gamma = \min\{ax | x \in k\text{ECSP}(G') \text{ and } x(e) = 1\}$. Moreover, if $ax \geq \alpha$ is facet defining for kECSP(G), then inequality (2.12) is also facet defining for kECSP(G'). In addition, if edges $e_1, \ldots, e_{k-1}, e_k, \ldots, e_t$ are added to G in this order and $a(e_k)$ is the lifting coefficient of e_k with respect to this order, *then* $a(e_k) \leq a'(e_k)$ *where* $a'(e_k)$ *is the lifting coefficient of* e_k *in any order* $e_{i_1}, \ldots, e_{i_{k-1}}, \ldots, e_{i_t}$ *such that* $i_l = l$ *for* $l = 1, \ldots, k-1$ *and* $i_s = k$ *for some* $s \geq k$.

Theorem 2.4. *Let* $G = (V,E)$ *be a graph and* $\pi = (W_1, W_2, V_1, \ldots, V_{2p})$, $p \geq 2$, *a partition of* V *which induces an odd path configuration. Let* C, I_1, *and* I_2 *be defined as in Section 2.1. Let* $U_1 = \bigcup_{i \in I_1} V_i$, $U_2 = \bigcup_{i \in I_2} V_i$, *and* $W = U_2 \cup V_{2p} \cup W_2$ *(respectively* $W = U_2 \cup W_2$*) if* p *is odd (respectively even). Suppose that conditions* 1–3 *of Theorem 2.3 hold. If* $G' = (V, E \cup L)$ *is a graph obtained from* G *by adding an edge set* L*, then the following inequality*

$$x(C) + \sum_{e \in L} a(e)x(e) \geq p, \tag{2.13}$$

with

$$a(e) = \begin{cases} 1 \textit{ if } e \in \left(\bigcup_{j=1,2} [W_j, U_1 \cup U_2] \right) \cup [W_1, W_2] \cup \left(\bigcup_{j=1,2p} [V_j, U_1 \cup U_2] \right) \textit{ or} \\ \quad e \in ([V_1, V_{2p} \cup W_2] \cup [V_{2p}, W_1 \cup W_2]) \cap \delta(W), \\ 2 \textit{ if } e \in [V_i, V_j], i,j \in \{2, \ldots, 2p-1\} \textit{ with } j > i+1 \textit{ and } i \textit{ even}, j \textit{ odd}, \\ \lambda \textit{ if } e \in [V_i, V_j] \textit{ with } i,j \in \{2, \ldots, 2p-1\}, j > i+1 \textit{ and } i \textit{ odd} \\ \quad \textit{or } i \textit{ and } j \textit{ have same parity}, \\ 0 \textit{ otherwise}, \end{cases}$$

where $1 \leq \lambda \leq 2$ *is the lifting coefficient obtained using the lifting procedure of Lemma 2.4, is facet defining for* $kECSP(G')$.

Proof. See Appendix. ∎

Observe that the lifting coefficients of the edges other than those between two subsets V_i and V_j such that $i,j \in \{2, \ldots, 2p-1\}$, $j > i+1$, i is odd or i and j have the same parity do not depend on the order of their addition in G. Inequalities (2.13) will be called *lifted odd path inequalities*. As it will turn out, these inequalities are very useful for our Branch-and-Cut algorithm.

2.3. *F*-partition inequalities

In [32], Mahjoub introduced a class of valid inequalities for 2ECSP(G) as follows. Let $(V_0, V_1, \ldots, V_p)$, $p \geq 2$, be a partition of V and $F \subseteq \delta(V_0)$ with $|F|$ odd. By adding the inequalities

$$x(\delta(V_i)) \geq 2 \quad \text{for } i = 1, \ldots, p, \tag{2.14}$$

$$-x(e) \geq -1 \quad \text{for } e \in F, \tag{2.15}$$

$$x(e) \geq 0 \quad \text{for } e \in \delta(V_0) \setminus F, \tag{2.16}$$

we obtain $2x(\Delta) \geq 2p - |F|$ where $\Delta = \delta(V_0, V_1, \ldots, V_p) \setminus F$. Dividing by 2 and rounding up the right hand side lead to

$$x(\Delta) \geq p - \frac{|F| - 1}{2}. \tag{2.17}$$

Inequalities (2.17) are called *F-partition inequalities*. Didi Biha [12] extended these inequalities for all $k \geq 2$. He showed that, given a partition $(V_0, V_1, \ldots, V_p)$, $p \geq 2$, of V and $F \subseteq \delta(V_0)$ with $F \neq \emptyset$, the inequality

$$x(\delta(V_0, V_1, \ldots, V_p) \setminus F) \geq \left\lceil \frac{kp - |F|}{2} \right\rceil, \tag{2.18}$$

is valid for kECSP(G). Note here that $|F|$ can be either odd or even. Also note that if kp and $|F|$ have the same parity, then the corresponding inequality (2.18) is implied by the cut and the trivial inequalities.

In what follows, we describe sufficient conditions for inequalities (2.18) to be facet defining. Theorems 2.5 and 2.6 describe these conditions for k odd and k even, respectively. Note that all the indices we will consider here will be modulo $2l+1$.

Theorem 2.5. *Let* $G = (V,E)$ *be a graph and* $k \geq 3$ *an odd integer. Let* $\pi = (W, V_1, \ldots, V_{2l+1}, U_1, \ldots, U_{2l+1})$, *with* $l \geq \frac{k-1}{2}$, *be a partition of* V *such that*

1. $G[W]$, $G[V_i]$, $G[U_i]$, $i = 1, \ldots, 2l+1$, *are* $(k+1)$*-edge connected,*
2. $|[W, V_i]| \geq k-2$ *for* $i = 1, \ldots, 2l+1$,
3. $|[U_i, U_{i+1}]| \geq \frac{k-1}{2}$ *for* $i = 1, \ldots, 2l+1$,
4. $|[V_i, V_{i+1}]| \geq 1$ *for* $i = 1, \ldots, 2l+1$,
5. $|[V_i, U_i]| \geq 1$ *and* $|[V_i, U_{i-1}]| \geq 1$ *for* $i = 1, \ldots, 2l+1$ *(see Fig. 2 for an illustration with* $k = 5$ *and* $l = 2$*).*

Let F_i *be an edge subset of* $[W, V_i]$ *such that* $|F_i| = k-2$, $i = 1, \ldots, 2l+1$ *and let* $F = \bigcup_{i=1}^{2l+1} F_i$. *Then the F-partition inequality*

$$x(\delta(\pi) \setminus F) \geq l(k+2) + \left\lceil \frac{k}{2} \right\rceil + 1, \tag{2.19}$$

induced by π *and* F*, defines a facet of* $kECSP(G)$.

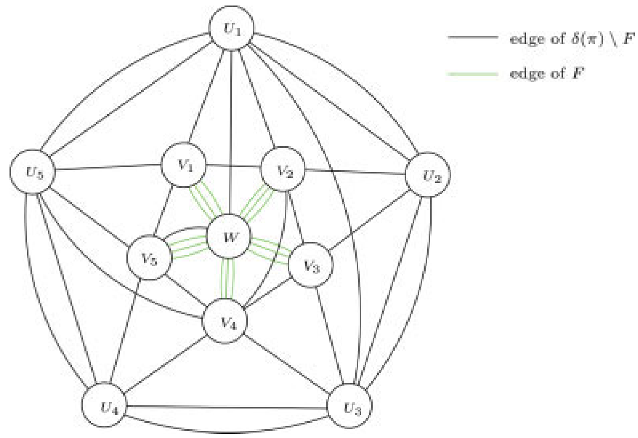

FIG. 2. An F-partition configuration with $k = 5$. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Proof. See Appendix. ∎

We now describe special cases in which inequalities (2.18) define facets when k is even. Consider a graph $G = (V, E)$ and an even integer $k = 2q$ with $q \geq 1$, a *generalized odd-wheel configuration* is given by an integer $l \geq 1$, a set of positive integers $\{p_1, \ldots, p_{2l+1}\}$ and a partition $\pi = (V_0, V_i^s, i = 1, \ldots, 2l+1, s = 0, \ldots, p_i)$ of V such that

1. $G[V_0]$ and $G[V_i^s]$ are $(k+1)$-edge connected, for $s = 1, \ldots, p_i$ and $i = 1, \ldots, 2l+1$,
2. $|[V_i^0, V_{i+1}^0]| \geq 2q$ for $i = 1, \ldots, 2l+1$,
3. $|[V_i^s, V_i^{s+1}]| \geq 2q$ for $s = 0, \ldots, p_i$ and $i = 1, \ldots, 2l+1$,
4. $[V_i^s, V_i^t] = \emptyset$ for $s, t \in \{1, \ldots, p_i\}$, $|s - t| > 1$ and $(s, t) \neq (0, p_i + 1)$, and $i = 1, \ldots, 2l+1$,
5. $[V_i^s, V_t^t] = \emptyset$ for $s \in \{1, \ldots, p_i\}$, $t \in \{1, \ldots, p_t\}$, $i, t \in \{1, \ldots, 2l+1\}$, $i \neq t$ (see Fig. 3).

Let F_i^0 be an edge subset of $[V_0, V_i^{p_i}]$ of q (respectively $q-1$) edges if q is odd (respectively even) and $F = \bigcup_{i=1}^{2l+1} F_i^0$.

With a generalized odd-wheel configuration with q odd (respectively even) we associate the following F-partition inequality induced by the partition π and F,

$$x(\delta(\pi) \setminus F) \geq q \sum_{i=1}^{2l+1} p_i + ql + \frac{q+1}{2},$$

$$\left(\text{respectively } x(\delta(\pi) \setminus F) \geq q \sum_{i=1}^{2l+1} p_i + (q+1)l + \frac{q+2}{2}\right). \tag{2.20}$$

Inequalities of type (2.20) will be called *generalized odd-wheel inequalities*. We have the following theorem given without proof, since it follows the same line as that of Theorem 2.5.

Theorem 2.6. *Inequalities* (2.20) *define facets of kECSP*(G).

2.4. SP-Partition Inequalities

In [7], Chopra introduces a class of valid inequalities for the kECSP when the graph G is outerplanar, k is odd, and each edge can be used more than once. Let $G = (V, E)$ be an outerplanar graph and $k \geq 1$ an odd integer. He showed that if $\pi = (V_1, \ldots, V_p)$, $p \geq 2$, is a partition of V, then the inequality

$$x(\delta(V_1, \ldots, V_p)) \geq \left\lceil \frac{k}{2} \right\rceil p - 1, \tag{2.21}$$

is valid for kECSP(G).

Didi Biha and Mahjoub [14] extended this result for general graphs and when each edge can be used at most once. They showed that if G is a graph and $\pi = (V_1, \ldots, V_p)$, $p \geq 2$, is a partition of V such that G_π is series-parallel, then inequality (2.21) is valid for kECSP(G). They called inequalities (2.21) *SP-partition inequalities*. They also described necessary conditions for inequality (2.21) to be facet defining, and showed that if G is series-parallel and k is odd, then kECSP(G) is defined by the trivial, cut and SP-partition inequalities. Further necessary conditions for inequalities (2.21) to be facet defining are given in [13]. In particular, Diarrassouba and Slama [13] showed the following.

Theorem 2.7 [13]. *Let $G = (V, E)$ be a $(k+1)$-edge connected graph and $k \geq 3$ an odd integer. Let $\pi = (V_1, \ldots, V_p)$, $p \geq 2$, be a partition of V such that G_π is series-parallel. If the SP-partition inequality induced by π defines a facet of kECSP(G) different from the trivial inequalities then*

1. *$|[V_i, V_{i+1}]| \geq \lceil \frac{k}{2} \rceil$ for $i = 1, \ldots, p$,*
2. *G_π is outerplanar.*

Note that the indices are taken modulo p. The following theorem gives some sufficient conditions for inequalities (2.21) to be facet defining.

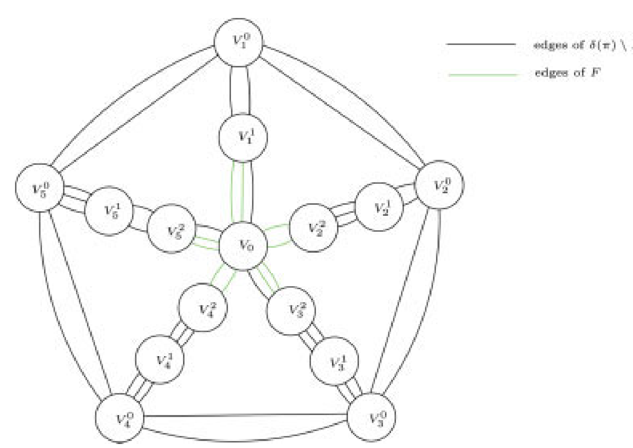

FIG. 3. A generalized odd-wheel configuration with $k = 4$. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

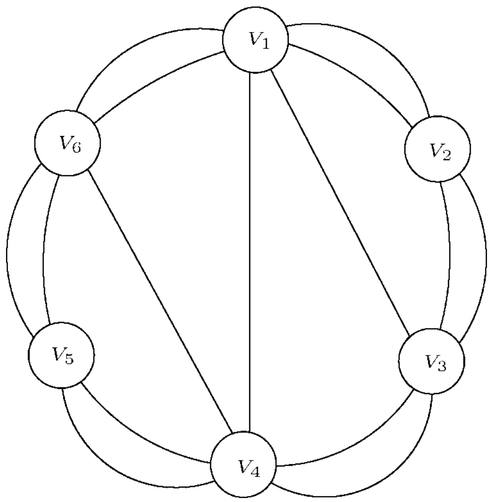

FIG. 4. An outerplanar configuration with $k = 3$.

Theorem 2.8. *Let $G = (V, E)$ be a graph and $k \geq 3$ an odd integer. Let $\pi = (V_1, \ldots, V_p)$, $p \geq 2$, be a partition of V such that G_π is outerplanar. Then the SP-partition inequality induced by π is facet defining for kECSP(G) if the following conditions hold*

1. *$G[V_i]$ is $(k+1)$-edge connected for $i = 1, \ldots, p$,*
2. *$|[V_i, V_{i+1}]| \geq \lceil \frac{k}{2} \rceil$, $i = 1, \ldots, p$*
   (see Fig. 4 for an illustration with $k = 3$).

Proof. See Appendix. ∎

Chopra [7] described a lifting procedure for inequalities (2.21) which can be presented as follows. Let $G = (V, E)$ be a graph and $k \geq 3$ an odd integer. Let $G' = (V, E \cup L)$ be a graph obtained from G by adding an edge set L. Let $\pi = (V_1, \ldots, V_p)$ be a partition of V such that G_π is series-parallel. Then the following inequality is valid for kECSP(G')

$$x(\delta_G(V_1, \ldots, V_p)) + \sum_{e \in L \cap \delta_{G'}(V_1, \ldots, V_p)} a(e)x(e) \geq \left\lceil \frac{k}{2} \right\rceil p - 1, \tag{2.22}$$

where $a(e)$ is the length (in terms of edges) of a shortest path in G_π between the endnodes of e, for all $e \in L \cap \delta_{G'}(V_1, \ldots, V_p)$.

We will call inequalities of type (2.22) *lifted SP-partition inequalities*. Chopra [7] also showed that, when G is outerplanar, inequality (2.22) defines a facet of kECSP(G') if G is maximal outerplanar, that is to say G is outerplanar and if we add a new edge in G the new graph is not outerplanar. This procedure can be easily extended to the case when each edge can be used at most once.

2.5. Partition Inequalities

In this section, we present a further class of inequalities, valid for kECSP(G), introduced by Grötschel et al. in [25], that generalizes the cut inequalities. These inequalities, called *partition inequalities*, are defined as follows.

Let $\pi = (V_1, \ldots, V_p)$, $p \geq 3$, be a partition of V. The *partition inequality* induced by π is given by

$$x(\delta(V_1, \ldots, V_p)) \geq \left\lceil \frac{kp}{2} \right\rceil. \tag{2.23}$$

If kp is even, then inequality (2.23) is redundant with respect to the cut inequalities. Grötschel et al. [25] gave sufficient conditions for the partition inequalities (2.23) to be facet defining.

Note that the partition inequalities are not a special case of the F-partition inequalities. In fact, if we consider a partition $\pi = (V_0, V_1, \ldots, V_p)$, $p \geq 2$, the partition inequality induced by π is

$$x(\delta(V_0, V_1, \ldots, V_p)) \geq \left\lceil \frac{k(p+1)}{2} \right\rceil. \tag{2.24}$$

However, the F-partition inequality induced by π and $F = \emptyset$ is given by

$$x(\delta(V_0, V_1, \ldots, V_p)) \geq \left\lceil \frac{kp}{2} \right\rceil. \tag{2.25}$$

One can remark that inequality (2.24) dominates inequality (2.25).

3. REDUCTION OPERATIONS

In this section, we are going to describe some graph reduction operations which will be utile for our Branch-and-Cut algorithm. These operations are based on the concept of critical extreme points of $P(G, k)$ introduced by Fonlupt and Mahjoub [18] for $k = 2$ and extended by Didi Biha and Mahjoub [16] for $k \geq 3$.

Before describing these operations, we shall first introduce some notation and definition. Let $G = (V, E)$ be a graph and $k \geq 2$ an integer. If $\bar{x}$ is a solution of $P(G, k)$, we will denote by $E_0(\bar{x})$, $E_1(\bar{x})$ and $E_f(\bar{x})$ the sets of edges $e \in E$ such that $\bar{x}(e) = 0$, $\bar{x}(e) = 1$ and $0 < \bar{x}(e) < 1$, respectively. We also denote by $C_d(\bar{x})$ the set of degree tight cuts $\delta(u)$ such that $\delta(u) \cap E_f(\bar{x}) \neq \emptyset$, and by $C_p(\bar{x})$ the set of proper tight cuts $\delta(W)$ with $\delta(W) \cap E_f(\bar{x}) \neq \emptyset$. Let $\bar{x}$ be an extreme point of $P(G, k)$. Thus there is a set of cuts $C_p^*(\bar{x}) \subseteq C_p(\bar{x})$ such that $\bar{x}$ is the unique solution of the system

$$S(\bar{x}) \begin{cases} x(e) = 0 & \text{for all } e \in E_0(\bar{x}); \\ x(e) = 1 & \text{for all } e \in E_1(\bar{x}); \\ x(\delta(u)) = k & \text{for all } \delta(u) \in C_d(\bar{x}); \\ x(\delta(W)) = k & \text{for all } \delta(W) \in C_p^*(\bar{x}). \end{cases}$$

Note that the system $S(\bar{x})$ cannot contain an equation $x(\delta(W)) = k$ such that $\delta(W) \cap E_f(\bar{x}) = \emptyset$. Such an equation is redundant with respect to $x(e) = 0$, $e \in E_0(\bar{x})$, and $x(e) = 1$, $e \in E_1(\bar{x})$.

Suppose that $\bar{x}$ is fractional. Let $\bar{x}'$ be a solution obtained by replacing some (but at least one) fractional components of $\bar{x}$ by 0 or 1 (and keeping all the other components of $\bar{x}$ unchanged). If $\bar{x}'$ is a point of $P(G,k)$, then it can be written as a convex combination of extreme points of $P(G,k)$. If $\bar{y}$ is such an extreme point, then $\bar{y}$ is said to be *dominated* by $\bar{x}$, and we write $\bar{x} \succ \bar{y}$. Note that if $\bar{x}$ dominates $\bar{y}$, then $\{e \in E \mid 0 < \bar{y}(e) < 1\} \subset \{e \in E \mid 0 < \bar{x}(e) < 1\}$, $\{e \in E \mid \bar{x}(e) = 0\} \subseteq \{e \in E \mid \bar{y}(e) = 0\}$ and $\{e \in E \mid \bar{x}(e) = 1\} \subseteq \{e \in E \mid \bar{y}(e) = 1\}$. The relation $\succ$ defines a partial ordering on the extreme points of $P(G,k)$. 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 $P(G,k)$. The minimal extreme points of $P(G,k)$ are called extreme points of *rank* 0. An extreme point x is said to be of *rank* p, if x only dominates extreme points of rank $\leq p-1$ and if it dominates at least one extreme point of rank $p-1$. We notice that if $\bar{x}$ is an extreme point of rank 1 and if we replace one fractional component of $\bar{x}$ by 1, keeping unchanged the other integral components, we obtain a feasible solution $\bar{x}'$ of $P(G,k)$ which can be written as a convex combination of integer extreme points of $P(G,k)$.

Didi Biha and Mahjoub [16] introduced the following reduction operations with respect to a solution $\bar{x}$ of $P(G,k)$.

- θ_1: delete an edge $e \in E$ such that $\bar{x}(e) = 0$;
- θ_2: contract a node subset $W \subseteq V$ such that $G[W]$ is k-edge connected and $\bar{x}(e) = 1$ for all $e \in E(W)$;
- θ_3: contract a node subset $W \subseteq V$ such that $|W| \geq 2$, $|\bar{W}| \geq 2$, $|\delta(W)| = k$, and $E(\bar{W})$ contains at least one edge with fractional value;
- θ_4: contract a node subset $W \subseteq V$ such that $|W| \geq 2$, $|\bar{W}| \geq 2$, $G[W]$ is $\lceil \frac{k}{2} \rceil$-edge connected, $|\delta(W)| = k+1$, and $\bar{x}(e) = 1$ for all $e \in E(W)$.

Starting from a graph G and a solution $\bar{x} \in P(G,k)$ and applying θ_1, θ_2, θ_3, θ_4, we obtain a reduced graph G' and a solution $\bar{x}' \in P(G',k)$. Didi Biha and Mahjoub [16] showed that $\bar{x}'$ is an extreme point of $P(G',k)$ if and only if $\bar{x}$ is an extreme point of $P(G,k)$. Moreover, they showed the following results.

Lemma 3.1. [16] *$\bar{x}'$ is an extreme point of rank 1 of $P(G',k)$ if and only if $\bar{x}$ is an extreme point of rank 1 of $P(G,k)$.*

Lemma 3.2. [16] *If $C_p^*(\bar{x}) = \emptyset$, then the graph induced by $E_f(\bar{x})$ is an odd cycle $C \subseteq E$ such that*

1. *$\bar{x}(e) = \frac{1}{2}$ for all $e \in C$,*
2. *$\bar{x}(\delta(u)) = k$ for all $u \in V(C)$.*

An extreme point $\bar{x}$ of $P(G,k)$ will be said *critical* if it is of rank 1 and none of the operations θ_1, θ_2, θ_3, θ_4 can be applied to it. If such an extreme point satisfies the assumption of Lemma 3.2, then it violates the following F-partition inequality

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

Hence, the critical extreme points of $P(G,k)$ that satisfy the assumption of Lemma 3.2 can be separated in polynomial time.

We will use operations θ_1, θ_2, θ_3, θ_4 in our Branch-and-Cut algorithm for the kECSP. As we will see, we use them as a preprocessing for the separation procedures.

4. A BRANCH-AND-CUT ALGORITHM FOR THE *k*ECSP

In this section, we describe a Branch-and-Cut algorithm for the kECSP. Our aim is to address the algorithmic applications of the theoritical results presented in the previous sections and describe some strategic choices made in order to solve that problem. So, let us assume that we are given a graph $G = (V,E)$ and a weight vector $w \in \mathbb{R}^E$ associated with the edges of G. Let $k \geq 3$ be the connectivity requirement for each node of V.

Given a fractional solution $\bar{x}$ of $P(G,k)$, we let $G' = (V',E')$ and $\bar{x}'$ be obtained by repeated applications of operations θ_1, θ_2, θ_3, θ_4 with respect to $\bar{x}$. As pointed out above, $\bar{x}'$ is an extreme point of $P(G',k)$ if and only if $\bar{x}$ is an extreme point of $P(G,k)$. Moreover, we have the following lemmas which can be easily seen.

Lemma 4.1. *Let $a'x \geq \alpha$ be an F-partition inequality (respectively partition inequality) valid for kECSP(G') induced by a partition $\pi' = (V_0', V_1', \ldots, V_p')$, $p \geq 2$, (respectively $\pi' = (V_1', \ldots, V_p')$, $p \geq 3$) of V'. Let $\pi = (V_0, V_1, \ldots, V_p)$, $p \geq 2$, (respectively $\pi = (V_1, \ldots, V_p)$, $p \geq 3$) be the partition of V obtained by expanding the subsets V_i' of π'. Let $ax \geq \alpha$ be an inequality such that*

$$a(e) = \begin{cases} a'(e) & \text{for all } e \in E', \\ 1 & \text{for all } e \in (E \setminus E') \cap \delta_G(\pi), \\ 0 & \text{otherwise.} \end{cases}$$

Then $ax \geq \alpha$ is valid for kECSP(G). Moreover, if $a'x \geq \alpha$ is violated by $\bar{x}'$, then $ax \geq \alpha$ is violated by $\bar{x}$.

Lemma 4.2. *Let $a'x \geq \alpha$ be an odd path inequality (respectively SP-partition inequality) valid for kECSP(G') induced by a partition $\pi' = (W_1', W_2', V_1', \ldots, V_{2p}')$, $p \geq 2$ (respectively $\pi = (V_1', \ldots, V_p')$, $p \geq 3$). Let $\pi = (W_1, W_2, V_1, \ldots, V_{2p})$, $p \geq 2$ (respectively $\pi = (V_1, \ldots, V_p)$, $p \geq 3$), be the partition of V obtained by expanding the elements of π'. Let $ax \geq \alpha$ be the corresponding lifted odd path inequality (respectively lifted SP-partition inequality) obtained from $a'x \geq \alpha$ by application of the lifting procedure described in Section 2.2 (respectively Section 2.4) for the edges of $E \setminus E'$. Then $ax \geq \alpha$ is violated by $\bar{x}$, if $a'x \geq \alpha$ is violated by $\bar{x}'$.*

Lemmas 4.1 and 4.2 show that looking for an odd path, F-partition, SP-partition, or a partition inequality violated by $\bar{x}$ reduces to looking for such inequality violated by $\bar{x}'$ on G'. Note that this procedure can be applied for any solution of

$P(G,k)$ and may, in consequence, permit to separate fractional solutions which are not necessarily extreme points of $P(G,k)$. In consequence, for more efficiency, our separation procedures will be performed on the reduced graph G'. The violated inequalities generated in G' with respect to $\bar{x}'$ are lifted to violated inequalities in G with respect to $\bar{x}$ using Lemmas 4.1 and 4.2.

We now describe the framework of our algorithm. To start the optimization we consider the following linear program given by the degree cuts associated with the vertices of the graph G together with the trivial inequalities, that is

$$\text{Min} \sum_{e \in E} w(e)x(e)$$
$$x(\delta(u)) \geq k \quad \text{for all } u \in V,$$
$$0 \leq x(e) \leq 1 \quad \text{for all } e \in E.$$

The optimal solution $\bar{y} \in \mathbb{R}^E$ of this relaxation of the kECSP is feasible for the problem if $\bar{y}$ is an integer vector that satisfies all the cut inequalities. Usually, the solution $\bar{y}$ is not feasible for the kECSP, and thus in each iteration of the Branch-and-Cut algorithm, it is necessary to generate further inequalities that are valid for the kECSP but violated by the current solution $\bar{y}$. For this, one has to solve the so-called *separation problem*. This consists, given a class of inequalities, in deciding whether the current solution $\bar{y}$ statisfies all the inequalities of this class, and if not, in finding an inequality that is violated by $\bar{y}$. An algorithm solving this problem is called a *separation algorithm*. The Branch-and-Cut algorithm uses the inequalities previously described and their separations are performed in the following order

1. cut inequalities,
2. *SP*-partition inequalities,
3. odd path inequalities,
4. *F*-partition inequalities,
5. partition inequalities.

We remark that all inequalities are global (i.e., valid for all the Branch-and-Cut tree) and several inequalities may be added at each iteration. Moreover, we go to the next class of inequalities only if we haven't found any violated inequalities in the current class. Our strategy is to try to detect violated inequalities at each node of the Branch-and-Cut tree to obtain the best possible lower bound and thus limit the number of generated nodes. Generated inequalities are added by sets of 200 or fewer at a time.

Now we describe the separation procedures used in our Branch-and-Cut algorithm. These are all heuristic procedures except that for the cut inequalities which is performed using an exact polynomial-time algorithm. The procedures are applied on G' with weights $(\bar{y}'(e), e \in E')$ associated with its edges where $\bar{y}'$ is the restriction on E' of the current LP-solution $\bar{y}$ (G' and $\bar{y}'$ are obtained by repeated applications of operations $\theta_1, \theta_2, \theta_3, \theta_4$).

The separation of the cut inequalities (1.3) can be performed by computing minimum cuts in G'. This can be done in polynomial time using Gusfield algorithm [27]. This algorithm produces the so-called *Gomory-Hu tree* [23] with the property that for all pairs of nodes $s, t \in V'$, the minimum (s,t)-cut in the tree is also a minimum (s,t)-cut in the graph G'. The algorithm requires $|V'| - 1$ maximum flow computations. The maximum flow computations are handled by the efficient Goldberg and Tarjan algorithm [22] that runs in $O(m'n' \log \frac{n'^2}{m'})$ time where m' and n' are the number of edges and nodes of G', respectively. Thus our separation algorithm for the cut inequalities is exact and runs in $O(m'n'^2 \log \frac{n'^2}{m'})$ time.

In what follows, we consider the separation of the odd path inequalities (2.1). For this, we need the following lemma.

Lemma 4.3. *Let $x \in \mathbb{R}^E$ be a fractional solution of $P(G,k)$ and $\pi = (W_1, W_2, V_1, \ldots, V_{2p})$, $p \geq 2$, a partition of V, which induces an odd path configuration. If each edge set $[V_i, V_{i+1}]$, $i = 1, \ldots, 2p-1$, contains an edge with fractional value and*

$$x([V_{i-1}, V_i]) + x([V_i, V_{i+1}]) \leq 1 \textit{ for } i = 2, \ldots, 2p-1,$$

then the odd path inequality induced by π is violated by x.

Proof. As $x([V_{i-1}, V_i]) + x([V_i, V_{i+1}]) \leq 1$, $i = 2, \ldots, 2p-1$, we have that

$$x([V_{2s-1}, V_{2s}]) + x([V_{2s}, V_{2s+1}]) \leq 1 \text{ for } s = 1, \ldots, p-1, \quad (4.1)$$

$$x([V_{2s}, V_{2s+1}]) + x([V_{2s+1}, V_{2s+2}]) \leq 1 \text{ for } s = 1, \ldots, p-1. \quad (4.2)$$

By multiplying inequality (4.1) by $\frac{p-s}{p}$ and inequality (4.2) by $\frac{s}{p}$ and summing the resulting inequalities, we obtain

$$\sum_{i \in I} x([V_i, V_{i+1}]) + \sum_{i \in \bar{I}} \frac{p-1}{p} x([V_i, V_{i+1}]) \leq p-1, \quad (4.3)$$

where $I = \{2, 4, 6, \ldots, 2p-2\}$ and $\bar{I} = \{1, 2, \ldots, 2p-1\} \setminus I$. Because each set $[V_i, V_{i+1}]$, $i = 1, \ldots, 2p-1$, contains an edge with fractional value, we have that $x([V_i, V_{i+1}]) < 1$ for all $i \in \bar{I}$. Hence

$$\sum_{i \in \bar{I}} x([V_i, V_{i+1}]) < p. \quad (4.4)$$

By multiplying inequality (4.4) by $\frac{1}{p}$ and summing the resulting inequality and inequality (4.3), we obtain

$$\sum_{i=1}^{2p-1} x([V_i, V_{i+1}]) < p,$$

and the result follows. ∎

Our separation heuristic is based on Lemma 4.3. The idea is to find a partition $\pi = (W'_1, W'_2, V'_1, \ldots, V'_{2p})$, $p \geq 2$, which induces an odd path configuration that satisfies the conditions of Lemma 4.3. The procedure works as follows. We first look, using a greedy method, for a path $\Gamma = \{e_1, \ldots, e_{2p-1}\}, p \geq 2$, in G' such that the edges $e_1, \ldots, e_{2p-1}$ have fractional values and $\bar{y}'(e_{i-1}) + \bar{y}'(e_i) \leq 1$, for $i = 2, \ldots, 2p-1$. If $v'_1, \ldots, v'_{2p}$ are the nodes of Γ taken in this order when going through Γ, we let $V'_i = \{v'_i\}$, $i = 1, \ldots 2p$, and $T_1 = (\bigcup_{i \in I_1} V'_i) \cup V'_1$ (respectively $T_1 = (\bigcup_{i \in I_1} V'_i) \cup V'_1 \cup V'_{2p}$) if p is odd (respectively even), and $T_2 = (\bigcup_{i \in I_2} V'_i) \cup V'_{2p}$ (respectively $T_2 = (\bigcup_{i \in I_2} V'_i)$) if p is odd (respectively even) where I_1 and I_2 are as defined in Section 2.1. In order to determine W'_1 and W'_2, we compute a minimum cut separating T_1 and T_2. If $\delta(W)$ is such a cut with $T_1 \subseteq W$, we let $W'_1 = W \setminus T_1$ and $W'_2 = V' \setminus (W \cup T_2)$. If the partition $\pi = (W'_1, W'_2, V'_1, \ldots, V'_{2p})$ thus obtained induces an odd path configuration, then, by Lemma 4.3, the corresponding odd path inequality is violated by $\bar{y}'$. If not, we apply again that procedure by looking for an other path. In order to avoid the detection of the same path, we label the edges of the detected paths so that they won't appear again when searching for a new path. This procedure is iterated until either a violated odd path inequality is found or all the edges, having fractional values, are labeled. The routine that permits to look for an odd path runs in $O(m'n')$ time. To compute the minimum cut separating T_1 and T_2, we use Goldberg and Tarjan algorithm [22]. Since this algorithm runs in $O(m'n' log \frac{n'^2}{m'})$ time, our procedure is implemented to run in $O(m'^2 n' \log \frac{n'^2}{m'})$ time.

In the lifting procedure for inequalities (2.1) given in Section 2.2 we have to compute a coefficient λ for some edges $e \in E \setminus E'$. As the computation of this coefficient is itself a hard problem, and $\lambda \leq 2$, we consider 2 as lifting coefficient for those edges rather than λ.

Now we discuss our separation procedure for the F-partition inequalities (2.18). These inequalities can be separated in polynomial time using the algorithm of Baïou et al. [2] when k is even and the edge set F is fixed. For the general case, we devised three heuristics to separate them.

Our first heuristic is based on Lemma 3.2. As pointed out by that lemma, if $\bar{x}$ is a critical extreme point of $P(G, k)$ such that $C^*_p(\bar{x}) = \emptyset$, then the edges having fractional values with respect to $\bar{x}$ have all a value equal to $\frac{1}{2}$ and form an odd cycle C. Moreover, $\bar{x}(\delta(u)) = k$ for all $u \in V(C)$ and

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

is an F-partition inequality violated by $\bar{x}$. The heuristic works as follows. It starts by determining an odd cycle in G' whose edges have fractional value and nodes are tight. Let $v'_1, \ldots, v'_p$, $p \geq 3$, be the nodes involved in this cycle. Then we let $V'_i = \{v'_i\}$, for $i = 1, \ldots, p$, and $V'_0 = V' \setminus \{v'_1, \ldots, v'_p\}$. We choose the edges of F among those of $\delta(V'_0)$ having values greater than $\frac{1}{2}$ and in such a way that $|F|$ and kp have different parities (if such an edge set F is empty then we look for an other partition). The cycle is obtained by a direct labeling procedure. Hence the heuristic runs in a linear time.

Before introducing our second heuristic, we first give the following lemma.

Lemma 4.4. *Let $x \in \mathbb{R}^E$ be a fractional solution of $P(G, k)$ and $\pi = (V_0, V_1, \ldots, V_p)$, $p \geq 2$, a partition of V such that $x(\delta(V_i)) = k$ for $i = 1, \ldots, p$. Then an F-partition inequality, induced by π and an edge set $F \subseteq \delta(V_0)$ such that $|F|$ and kp have different parities is violated by x if the following inequality holds*

$$|F| - x(F) + x(\delta(V_0) \setminus F) < 1. \tag{4.5}$$

Proof. As $x(\delta(V_i)) = k$, $i = 1, \ldots, p$, we have that

$$\sum_{i=1}^{p} x(\delta(V_i)) = 2x(\delta(V_1, \ldots, V_p)) + x(\delta(V_0)) = kp.$$

This together with (4.5) yield

$$-2x(F) + 2x(\delta(V_0)) + 2x(\delta(V_1, \ldots, V_p)) < kp - |F| + 1,$$

and thus the statement follows. ∎

The heuristic is based on Lemma 4.4. It starts by determining all the nodes u of V' such that $\bar{y}'(\delta(u)) = k$ and $\delta(u)$ contains at least one edge with fractional value. Let $\{v'_1, \ldots, v'_p\}$, $p \geq 2$, be the set of such nodes. We consider the partition $(V'_0, V'_1, \ldots, V'_p)$ such that $V'_i = \{v'_i\}$, for $i = 1, \ldots, p$, and $V'_0 = V' \setminus \{v_1, \ldots, v_p\}$, and choose the edges of F in a similar way as in the first heuristic. If inequality (4.5) holds with respect to F and V'_0, then by Lemma 4.4 the F-partition inequality corresponding to $(V'_0, V'_1, \ldots, V'_p)$ and F is violated by $\bar{y}'$.

Before presenting our last heuristic for the F-partition inequalities, let us first remark that a partition $(V'_0, V'_1, \ldots, V'_p)$ and an edge set $F \subseteq \delta(V'_0)$ may induce a violated F-partition inequality if $\bar{y}'(\delta(V'_0))$ is high and the edges of F are among those of $\delta(V'_0)$ with high values. Our heuristic tries to find such a partition. For this, we first compute a Gomory-Hu tree in G' with the weights $(1 - \bar{y}'(e), e \in E')$ associated with its edges. Then from each proper cut $\delta(W)$ with $V' \setminus W = \{v'_1, \ldots, v'_p\}$, $p \geq 2$, obtained from the Gomory-Hu tree, we consider the partition $\pi = (V'_0, V'_1, \ldots, V'_p)$ such that $V'_i = \{v'_i\}$, for $i = 1, \ldots, p$, and $V'_0 = W$. The edge set F is chosen in a similar way as in the previous heuristics. Since the computation of the Gomory-Hu tree can be done in $O(m'n'^2 \log \frac{n'^2}{m'})$ time, the heuristic runs in $O(m'n'^2 \log \frac{n'^2}{m'})$.

These three heuristics are applied in the Branch-and-Cut algorithm in that order.

Now we turn our attention to the separation of the SP-partition inequalities (2.21). These inequalities can be separated in polynomial time using the algorithm of Baöu et al. [2] when G' is series-parallel. That algorithm uses a reduction of the separation problem to the minimization of a submodular function. Recently, Didi Biha et al. [17] devised a pure combinatorial algorithm for the separation of the SP-partition inequalities when the graph is series-parallel. For our purpose, we devised a heuristic to separate inequalities (2.21) in the general case. This heuristic is based on Theorems 2.7 and 2.8. The main idea of the heuristic is to determine a partition $\pi = (V'_1, \ldots, V'_p)$, $p \geq 3$, of V' which induces an outerplanar graph such that $|[V'_i, V'_{i+1}]| \geq \left\lceil \frac{k}{2} \right\rceil$, $i = 1, \ldots, p$, (modulo p) (see Fig. 4), and for every consecutive sets V'_i and V'_j, the edge set $[V'_i, V'_j]$ contains at least one edge with fractional value. To this end, we look in G' for a path $\Gamma = \{v'_1v'_2, v'_2v'_3, \ldots, v'_{p-2}v'_{p-1}\}$, $p \geq 3$, such that $|[v'_i, v'_{i+1}]| \geq \left\lceil \frac{k}{2} \right\rceil$ and $[v'_i, v'_{i+1}]$ contains one edge or more with fractional value, for $i = 1, \ldots, p-2$. We then let $V'_i = \{v'_i\}$, $i = 1, \ldots, p-1$, and $V'_p = V' \setminus \{v'_1, \ldots, v'_{p-1}\}$. Afterward, we check by a simple heuristic if the graph G'_π is outerplanar. Finally, we check if the *SP*-partition inequality induced by π is violated by $\bar{y}'$ or not. If either the graph G'_π is not outerplanar or the *SP*-partition inequality, induced by π, is not violated by $\bar{y}'$, we apply again this procedure by looking for an other path. In order to avoid the detection of the same path, we label the nodes we met during the search of the previous ones, so that they would not be considered in the search of a new path. This process is iterated until either we find a violated *SP*-partition inequality or all the nodes of V' are labeled. The heuristic can be implemented to run in $O(m'n')$ time.

Now we discuss the separation of the partition inequalities (2.23). First observe that if $\pi = (V'_1, \ldots, V'_p)$ is a partition of V', with $p \geq 3$ and odd, such that $\bar{y}'(\delta(V'_i)) = k$, for $i = 1, \ldots, p$, then the partition inequality induced by π is violated by $\bar{y}'$. Thus one can devise a heuristic to separate inequalities (2.23) which consists in finding a partition $\pi = (V'_1, \ldots, V'_p)$, with $p \geq 3$ and odd, such that $\bar{y}'(\delta(V'_i))$ is as small as possible for $i = 1, \ldots, p$. To do this, we compute a Gomory-Hu tree, say $\mathcal{T}$, in G' with the weights $(\bar{y}'(e), e \in E')$ associated with its edges. After that, we contract the disjoint node subsets that induce proper tight cuts in $\mathcal{T}$. Let $V'_1, \ldots, V'_t$ be these sets and $\{v_{t+1}, \ldots, v_p\} = V' \setminus (\bigcup_{i=1}^{t} V'_i)$. We then consider the partition $(V'_1, \ldots, V'_t, \{v_{t+1}\}, \ldots, \{v_p\})$ and check whether or not the corresponding partition inequality is violated by $\bar{y}'$. This algorithm leads to an $O(m'n'^2 \log \frac{n'^2}{m'})$ time complexity.

To store the generated inequalities, we create a pool whose size increases dynamically. 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. We first separate inequalities from the pool. If all the inequalities in the pool are satisfied by the current LP-solution, we separate the classes of inequalities in the order given earlier.

Another important issue in the effectiveness of the Branch-and-Cut algorithm is the computation of a good upper bound at each node of the Branch-and-Cut tree. To do this, if the separation procedures do not generate any violated inequality and the current solution $\bar{y}$ is still fractional, then we transform $\bar{y}$ into a feasible solution of the *k*ECSP, say $\hat{\bar{y}}$, by rounding up to 1 all the fractional components of $\bar{y}$. We then try to reduce the weight of the solution thus obtained by removing from the subgraph $H = (V, \hat{E})$ induced by $\hat{\bar{y}}$ some unecessary edges, that is to say edges which do not affect the k-edge connectedness of H. To this end, we remove from $\hat{E}$ each edge $e = uv$ such that $|\delta(u) \cap \hat{E}| \geq k+1$ and $|\delta(v) \cap \hat{E}| \geq k+1$. We then check if the resulting edge set, say $\hat{E}'$, induces a k-edge connected subgraph of G by computing a Gomory-Hu tree. If there exists in $\hat{E}'$ a cut $\delta(W)$, $W \subseteq V$, containing less than k edges, then we add in $\hat{E}'$ edges of $[W, V \setminus W] \setminus \delta(W)$ that have been previously removed from $\hat{E}$ as many as necessary in order to satisfy the cut $\delta(W)$. We do this until the graph $(V, \hat{E}')$ becomes k-edge connected. Note that we add to each violated cut the edges having the smallest weights.

5. COMPUTATIONAL RESULTS

The Branch-and-Cut algorithm described in the previous section has been implemented in C++, using ABACUS 2.4 alpha [1,36] to manage the Branch-and-Cut tree, and CPLEX 9.0 [11] as LP-solver. It was tested on a Pentium IV 3.4 Ghz with 1 Gb of RAM, running under Linux. We fixed the maximum CPU time to 5 h. The test problems were obtained by taking TSP test problems from the TSPLIB library [37]. The test set consists in complete graphs whose edge weights are the rounded euclidian distance between the edge's vertices. The tests were performed for $k = 3, 4, 5$. In all our experiments, we have used the reduction operations described in the previous sections, unless otherwise specified. Each instance is given by its name followed by an extension representing the number of nodes of the graph. The other entries of the various tables are:

NCut : number of generated cut inequalities;
NSP : number of generated *SP*-partition inequalities;
NOP : number of generated odd path inequalities;
NFP : number of generated *F*-partition inequalities;
NP : number of generated partition inequalities;
COpt : weight of the optimal solution obtained;
Gap1 : the relative error between the best upper bound (the optimal solution if the problem has been solved to optimality) and the lower bound obtained at the root node of the Branch-and-Cut tree using only the cut and the trivial inequalities;
Gap2 : the relative error between the best upper bound (the optimal solution if the problem has been solved to optimality) and the lower bound obtained at the root node of the Branch-and-Cut tree;
NSub : number of subproblems in the Branch-and-Cut tree;
TT : total CPU time in hours:min:sec.

The instances indicated with "*" are those whose CPU time exceeded 5 h. For these instances, the gap is indicated in italic.

Our first series of experiments concerns the *k*ECSP for $k = 3$. The instances we have considered have graphs with 14 up to 318 nodes. The results are summarized in Table 1. It appears from Table 1 that all the instances have been solved to optimality within the time limit except the last five instances. Also we have that four instances (burma14, gr21, fri26, brazil58) have been solved in the cutting plane phase (i.e., no branching is needed). For most of the other instances, the relative error between the lower bound at the root node of the Branch-and-Cut tree and the best upper bound (Gap2) is

TABLE 1. Results for $k = 3$ with reduction operations.

Instance	NCut	NSP	NOP	NFP	NP	COpt	Gap1	Gap2	NSub	TT
burma14	4	3	0	0	4	5,530	4.67	0.00	1	0:00:01
ulysses16	5	7	1	15	7	11,412	1.17	0.39	3	0:00:11
gr21	5	6	1	0	2	4,740	1.65	0.00	1	0:00:01
fri26	9	5	0	0	0	1,543	1.30	0.00	1	0:00:01
bayg29	14	16	2	33	2	2,639	1.76	0.19	7	0:00:01
dantzig42	41	31	6	90	18	1,210	2.27	0.68	71	0:00:07
att48	34	34	5	60	9	17,499	1.83	0.56	61	0:00:06
berlin52	36	31	12	97	6	12,601	1.66	0.45	33	0:00:03
brazil58	46	42	2	36	29	42,527	2.67	0.00	1	0:00:05
eil76	9	12	3	298	2	876	0.63	0.06	7	0:00:03
pr76	130	207	72	2,231	54	187,283	3.9	1.50	6767	0:35:32
rat99	41	26	13	341	23	2,029	1.26	0.38	41	0:00:47
kroA100	170	197	31	1,207	57	36,337	4.64	0.97	4201	0:54:06
kroB100	130	114	37	830	47	37,179	2.61	0.73	723	0:08:00
rd100	101	74	11	418	18	13,284	1.91	0.43	171	0:03:37
eil101	86	72	21	3,604	15	1,016	1.06	0.55	1109	0:17:41
lin105	179	198	47	829	68	25,530	3.66	0.69	1031	0:22:39
pr107	201	190	34	674	114	70,852	2.48	0.84	2071	1:26:49
gr120	50	45	6	588	17	11,442	1.12	0.19	99	0:11:15
bier127	46	59	4	276	13	198,184	1.50	0.15	11	0:01:55
ch130	121	132	30	1,355	40	10,400	2.27	0.55	1693	1:05:05
ch150	92	93	19	588	22	11,027	2.04	0.41	193	0:20:31
kroA150	155	143	41	845	47	44,718	2.27	0.53	1205	1:16:35
kroB150	130	110	16	952	48	43,980	2.26	0.31	437	0:38:43
rat195	24	19	3	514	1	3,934	0.48	0.06	7	0:08:21
d198	171	105	23	617	59	25,624	2.00	0.21	159	1:04:19
gr202	77	69	14	558	22	65,729	1.02	0.11	69	0:13:16
*pr226	364	248	35	162	41	—	*11.05*	*9.02*	261	5:00:00
*gr229	179	245	23	1,568	94	—	*2.43*	*1.00*	1219	5:00:00
*pr264	275	181	145	668	62	—	*12.56*	*12.29*	69	5:00:00
*a280	142	84	56	2,539	59	—	*3.73*	*2.69*	459	5:00:00
*lin318	189	147	15	610	58	—	*6.5*	*4.94*	25	5:00:00

less than 1%. We also observe that our separation procedures detect a large enough number of *SP*-partition and *F*-partition inequalities and seem to be quite efficient.

Our second series of experiments concerns the kECSP with $k = 4, 5$. The results are given in Table 2 for $k = 4$ and Table 3 for $k = 5$. The instances considered have graphs with 52 up to 561 nodes. Note that for $k = 4$, the *SP*-partition and partition inequalities are redundant with respect to the cut inequalities (1.3). Thus these inequalities are not considered in the resolution process for $k = 4$, and therefore do not appear in Table 2.

First observe that for $k = 4$, the CPU time for all the instances is relatively small and most of the instances have been solved in less than 1 min. We can also observe that 23 instances over 27 are solved in the cutting plane phase. Moreover, a few number of odd path inequalities are generated. However a large enough number of *F*-partition inequalities is detected. Thus these latter inequalities seem to be very effective for solving the kECSP when k is even. This also shows that the kECSP is easier to solve when k is even, what is also confirmed by the results of Table 3 for $k = 5$. In fact, the instance pr264 has been solved for $k = 4$ in 1 s, whereas it could not be solved to optimality for $k = 5$ after 5 h. The same observation can be done for pr439. Also, we can remark that the CPU time for all the instances when $k = 5$ is higher than that when $k = 4$. For instance, the test problem d198 has been solved in 1 h 50 mn when $k = 5$, whereas only 16 s were needed to solve it for $k = 4$.

Compared with Tables 1, 2, and 3 also show that, for the same parity of k, the kECSP becomes easier to solve when k increases. In fact, with $k = 3$, we could not solve to optimality instances with more than 202 nodes, whereas for $k = 5$, we could solve larger instances.

The results for $k = 3, 4, 5$ can also be compared with those obtained by Kerivin et al. [30] for the 2ECSP. It turns out that for the same instances, the problem has been easier to solve for $k = 2$ than for $k = 3$. However, for $k = 4$ the problem appeared to be easier to solve than for $k = 2$. This shows again that the case when k is odd is harder to solve than that when k is even and that the problem becomes easier when k increases with the same parity.

To evaluate the impact of the reduction operations θ_1, θ_2, θ_3, θ_4 on the separation procedures, we tried to solve the kECSP, for $k = 3$, without using them. The results are given in Table 4.

As it appears from Tables 1 and 4, the CPU time increased for the majority of the instances when the reduction operations are not used. In particular, for the instance pr107, without the reduction operations, we could not reach the optimal solution after 5 h, whereas with the reduction operations,

TABLE 2. Results for $k = 4$.

Instance	NCut	NOP	NFP	COpt	Gap2	NSub	TT
berlin52	5	0	2	18, 295	0.00	1	0:00:01
pr76	3	0	4	266, 395	0.00	1	0:00:01
kroA100	10	0	11	51, 221	0.00	1	0:00:47
kroB100	9	5	123	53, 597	0.08	21	0:00:09
rd100	10	1	91	19, 130	0.00	1	0:00:05
eil101	0	0	60	1, 453	0.00	1	0:00:02
lin105	20	1	5	36, 353	0.00	1	0:00:01
pr107	29	0	0	98, 381	0.00	1	0:00:01
gr120	6	0	36	16, 400	0.00	1	0:00:02
bier127	16	0	0	282, 207	0.00	1	0:00:01
ch130	12	0	132	14, 854	0.00	1	0:00:05
ch150	12	2	70	15, 854	0.00	1	0:00:02
kroA150	13	0	27	64, 249	0.00	1	0:00:02
kroB150	20	0	4	62, 710	0.00	1	0:00:01
rat195	0	0	37	5, 750	0.00	1	0:00:13
d198	43	0	71	35, 404	0.01	3	0:00:16
gr202	13	3	220	94, 841	0.02	3	0:01:28
pr226	91	0	6	183, 537	0.00	1	0:00:04
gr229	24	2	15	318, 565	0.00	1	0:00:03
pr264	59	1	7	122, 941	0.00	1	0:00:06
a280	3	0	180	6, 317	0.00	1	0:01:00
pr299	30	0	427	117, 559	0.00	1	0:00:20
lin318	28	0	2	105, 000	0.00	1	0:00:06
rd400	21	2	232	36, 676	0.00	1	0:07:39
pr439	78	3	61	264, 975	0.02	19	0:02:52
si535	0	0	4	53, 604	0.00	1	0:00:39
pa561	10	1	306	6, 724	0.00	1	0:08:37

TABLE 3. Results for $k = 5$.

Instance	NCut	NSP	NOP	NFP	NP	COpt	Gap2	NSub	TT
berlin52	5	2	2	26	2	24,845	0.00	1	0:00:01
pr76	2	0	0	52	1	372,392	0.00	1	0:00:01
kroA100	5	1	5	76	6	71,422	0.04	11	0:00:06
kroB100	6	1	2	83	5	74,241	0.01	3	0:00:06
rd100	6	2	6	193	5	26,168	0.01	5	0:00:24
eil101	1	0	0	309	0	1,938	0.00	1	0:01:10
lin105	9	1	3	119	3	50,711	0.00	1	0:00:26
pr107	92	40	57	680	33	132,870	0.41	381	0:14:45
gr120	2	0	3	93	3	22,024	0.11	27	0:00:17
bier127	22	2	12	450	8	383,165	0.09	25	0:04:25
ch130	1	0	0	45	0	20,508	0.01	3	0:00:05
ch150	5	0	7	58	1	21,791	0.01	37	0:00:50
kroA150	9	0	5	141	3	87,950	0.07	11	0:00:19
kroB150	14	1	7	462	6	85,583	0.02	11	0:15:39
rat195	1	0	0	508	0	7,773	0.00	1	0:20:54
d198	56	9	6	1,093	32	47,614	0.15	337	1:50:40
gr202	0	0	0	64	0	128,990	0.00	1	0:00:31
pr226	142	34	20	661	50	260,878	0.58	103	2:38:50
gr229	18	1	11	679	9	434,422	0.06	43	0:31:58
*pr264	105	12	38	1,327	28	—	*1.78*	43	5:00:00
a280	2	0	2	302	0	8,643	0.02	7	0:05:05
pr299	11	3	2	637	1	161,576	0.00	1	0:05:12
lin318	24	3	11	1,548	11	144,341	0.02	7	4:34:39
rd400	11	1	15	691	6	49,893	0.01	17	1:29:09
*pr439	46	2	8	746	0	—	*3.46*	1	5:00:00
si535	0	0	0	0	0	79,115	0.00	1	0:00:19
pa561	1	0	2	286	1	9,161	0.00	1	3:26:58

TABLE 4. Results for $k = 3$ without reduction operations.

Instance	NCut	NSP	NOP	NFP	NP	COpt	Gap2	NSub	TT
berlin52	31	28	19	44	4	12,601	0.44	15	0:00:04
brazil58	50	27	1	28	31	42,527	0.22	3	0:00:07
eil76	9	6	3	102	2	876	0.00	1	0:00:01
pr76	103	168	65	1,378	37	187,283	1.60	3,483	0:38:46
rat99	41	19	10	223	17	2,029	0.32	61	0:01:29
kroA100	193	234	47	1,765	70	36,337	1.42	7,575	4:13:38
kroB100	141	142	36	899	38	37,179	0.98	1,337	0:45:34
rd100	103	84	15	445	21	13,284	0.40	233	0:11:40
eil101	77	58	26	2,527	12	1,016	0.38	801	0:18:50
lin105	161	158	50	569	53	25,530	0.61	547	0:34:25
*pr107	218	221	136	1,101	104	—	*0.81*	4,447	5:00:00
gr120	42	38	6	252	15	11,442	0.18	93	0:05:38
bier127	58	56	9	240	12	198,184	0.16	17	0:04:43
ch130	141	147	38	1,590	45	10,400	0.52	2,459	4:10:31
ch150	90	76	15	391	23	11,027	0.39	107	0:21:07
kroA150	155	135	23	705	56	44,718	0.55	1,107	3:08:37
kroB150	150	141	22	1,006	43	43,980	0.31	535	1:55:20
rat195	23	18	7	898	1	3,934	0.01	19	0:19:23
d198	192	118	25	720	50	25,624	0.27	585	5:03:16
gr202	73	62	13	278	23	65,729	0.05	37	0:37:31

it has been solved to optimality after 1 h 26 mn. Also, the CPU time for the instances ch130 and d198 increased from 1 h to more than 4 h. Moreover, we remark that when using the reduction operations, we generate more *SP*-partition, F-partition and partition inequalities and fewer nodes in the Branch-and-Cut tree. This implies that our separation heuristics are less efficient without the reduction operations. It seems then that the reduction operations play an important role in the resolution of the problem. They permit to strengthen much more the linear relaxation of the problem and accelerate its resolution.

We also tried to measure the effect of the different non-basic classes of inequalities (i.e., inequalities other than cut and trivial inequalities). For this, we have first considered a Branch-and-Cut algorithm for the kECSP with $k = 3$ using only the cut constraints in addition to the trivial ones. As it appears from Table 1, for all the instances we have that Gap1 is greater than Gap2. For example, for the instances KroA100 and rat195, the gap is increased by almost 3%.

Furthermore, in this case, we could not solve any of the instances with more than 52 nodes. Even more, after less than 10 min of CPU time, the Branch-and-Cut tree got a very big size and the resolution process stops. To illustrate this, take for example the instance brazil58. For this instance, the Branch-and-Cut tree contained 11,769 nodes after 10 min when the Branch-and-Cut algorithm used only the cut and trivial inequalities, whereas it has been solved without branching when using the other classes of inequalities.

Finally, we tried to evaluate separately the efficiency of each class of the non-basic inequalities. For this, we also considered the case when $k = 3$. We have seen that all the classes of inequalities have a big effect on the resolution of the problem. In particular, the *SP*-partition inequalities seem to play a central role. This can be seen by considering the instance d198. This instance has been solved in 1 h 04 mn using all the constraints. However, without the *SP*-partition inequalities, we could not reach the optimal solution after 5 h. We also remarked that the gap2 increased when one of these classes of inequalities is not used in the Branch-and-Cut algorithm.

6. CONCLUDING REMARKS

In this article, we have studied the k-edge connected subgraph problem with high connectivity requirement, that is, when $k \geq 3$. We have presented some classes of valid inequalities and described some conditions for these inequalities to be facet defining for the associated polytope. We also discussed separation heuristics for these inequalities. Using these results, we have devised a Branch-and-Cut algorithm for the problem. This algorithm uses some reduction operations.

Our computational results have shown that the odd path, the F-partition, the *SP*-partition, and the partition inequalities are very effective for the problem when k is odd. They have also shown the importance of the F-partition inequalities for the even case. We could also see the importance of our separation heuristics. In particular, our heuristics to separate the *SP*-partition and F-partition inequalities have appeared to be very efficient. In addition, the reduction operations have been essential for having a good performance of the Branch-and-Cut algorithm. In fact, they permitted to considerably reduce the size of the graph supporting a fractional solution and to accelerate the separation process.

These experiments also showed that the kECSP is easier to solve when k is even and that, for the same parity of k, the problem becomes easier to solve when k increases.

One of the separation heuristic devised for the F-partition inequalities is based on a partial characterization of the critical extreme points of the linear relaxation of the k-edge connected subgraph polytope. It would be very interesting to have a complete characterization of these points. This may yield the identification of new facet defining inequalities for the problem. It may also permit to devise more appropriate separation heuristics for the inequalities given in this paper.

In many real instances, we may consider node-connectivity instead of edge-connectivity. The study presented in this paper may be very usefull for the k-node connected subgraph problem for which we require k node-disjoint paths between every pair of nodes.

In addition to the survivability aspect, one can consider the capacity dimensioning of the network. These issues have been mostly treated separately in the literature. It would be interesting to extend the study developed in this paper to the more general capacitated survivable network design model.

Acknowledgments

The authors thank the anonymous referees for their valuable comments that permitted improvement to the presentation of the article.

APPENDIX A. PROOF OF THEOREM 2.4

Let us consider the following edge subsets of L:

$$L_1 = \left(\bigcup_{j=1,2} [W_j, U_1 \cup U_2] \right) \cup [W_1, W_2]$$
$$\cup \left(\bigcup_{j=1,2p} [V_j, U_1 \cup U_2] \right)$$
$$\cup (([V_1, V_{2p} \cup W_2] \cup [V_{2p}, W_1 \cup W_2]) \cap \delta(W)),$$
$$L_2 = \{[V_i, V_j], i, j \in \{2, \ldots, 2p-1\}, j > i+1, \; i \text{ even }, j \text{ odd}\},$$
$$L_3 = \{[V_i, V_j], i, j \in \{2, \ldots, 2p-1\}, j > i+1, \; i \text{ odd or, } i \text{ and } j \text{ have the same parity}\},$$
$$L_4 = L \setminus (L_1 \cup L_2 \cup L_3).$$

We will first show that the lifting coefficient of the edges of L_4 is equal to 0, independently of the order in which they are added to G. Let e be an edge of L_4 and let us denote by $a'x \geq \alpha'$ the lifted inequality obtained on G'. As, by our assumptions, (2.1) defines a facet of kECSP(G), $a'x \geq \alpha'$ also defines a facet of kECSP(G'). As $a'x \geq \alpha'$ is different from the trivial inequality $x(e) \geq 0$, there must exist a solution $F' \subseteq E'$ of the kECSP on G' such that $e \in F'$ and whose the incidence vector satisfies $a'x \geq \alpha'$ with equality. Let $h_1, \ldots h_k$ be the edges of E between V_1 and W_1. Note that $a'(h_1) = \cdots = a'(h_k) = 0$. We will distinguish two cases.

Case 1. $|[F' \cap \{h_1, \ldots, h_k\}]| \leq k-1$. Let h_i be an edge not contained in F'. Let $F'' = (F' \setminus \{e\}) \cup \{h_i\}$. Since F' induces a k-edge connected subgraph of G', F'' so is. Hence we have that $a'x^{F''} = a'x^{F'} - a'(e) + a'(h_i) \geq \alpha'$. This yields $a'(e) \leq a'(h_i)$. As $a'(h_i) = 0$, and by Lemma 2.1, $a'(e) \geq 0$, we get $a'(e) = 0$.

Case 2. $\{h_1, \ldots, h_k\} \subseteq F'$. Here we also have that $F'' = F' \setminus \{e\}$ induces a k-edge connected subgraph of G'. As $a'x^{F''} = a'x^{F'} - a'(e) \geq \alpha'$, and thus $a'(e) \leq 0$, it follows, by Lemma 2.1, that $a'(e) = 0$.

Therefore $a(e) = 0$ for all $e \in L_4$, and this, independently of the order in which e is added to G.

Now we consider the edges of $L \setminus L_4$. For this, we give the following claim.

Claim. *$a(e) \geq 1$ if $e \in L_1 \cup L_3$, and $a(e) \geq 2$ if $e \in L_2$.*

Proof. We will show first that if we add one edge $e \in L_1$ (respectively $e \in L_2$) (respectively $e \in L_3$) to G, the lifting coefficient of e in the new graph is 1 (respectively 2) (respectively 1). For this, let us denote by $\tilde{G} = (V, \tilde{E})$ the graph obtained by adding the edge e, that is, $\tilde{E} = E \cup \{e\}$. Suppose first that $e \in L_1$ and assume that, for instance, $e \in [W_{j_0}, V_{i_0}]$, with $i_0 \in \{2, \ldots, 2p-1\}$ and even, and $j_0 \in \{1, 2\}$ (if i_0 is odd, it suffices to consider the path $V_1, \ldots, V_{2p}$ in the opposite way). Note that any solution $\tilde{F} \subseteq \tilde{E}$ of the kECSP on $\tilde{G}$ must cover the node sets $V_2, \ldots, V_{i_0-1}$ and $V_{i_0+1}, \ldots, V_{2p-1}$ by edges from C. By Lemma 2.3, $\tilde{F}$ must use at least $\lceil \frac{i_0-2}{2} \rceil + \lceil \frac{2p-i_0-1}{2} \rceil = p-1$ edges from C. Thus $\gamma \geq p-1$ where γ is as defined in Lemma 2.4. Moreover, because the conditions of Theorem 2.3 are satisfied, by the claim given in the proof of that theorem, the edge set $\tilde{F}_1 = \{e_2, e_4, \ldots, e_{i_0-2}\} \cup \{e_{i_0+1}, e_{i_0+3}, \ldots, e_{2p-1}\} \cup \Gamma \cup \{e\}$ induces a k-edge connected subgraph of $\tilde{G}$. As $\tilde{F}_1$ contains e and uses exactly $p-1$ edges from C, we have that $\gamma = p-1$. By Lemma 2.4, it then follows that the lifting coefficient of e is equal to 1.

Consider now an edge $e \in L_2$ and suppose that $e \in [V_{i_0}, V_{j_0}]$ with $i_0, j_0 \in \{2, \ldots, 2p-1\}$, $j_0 > i_0 + 1$, and i_0 is even and j_0 odd. If $\tilde{F}$ is a solution of the kECSP on $\tilde{G}$, then $\tilde{F}$ must cover the node sets $V_2, \ldots, V_{i_0-1}$, $V_{i_0+1}, \ldots, V_{j_0-1}$ and $V_{j_0+1}, \ldots, V_{2p-1}$. Thus by Lemma 2.3, $\tilde{F}$ must use $\lceil \frac{i_0-2}{2} \rceil + \lceil \frac{j_0-i_0-1}{2} \rceil + \lceil \frac{2p-j_0-1}{2} \rceil = p-2$ edges from C. Thus, $\gamma \geq p-2$. Now let $\tilde{F}_2 = \{e_2, e_4, \ldots, e_{i_0-2}\} \cup \{e_{i_0+1}, e_{i_0+3}, \ldots, e_{j_0-2}\} \cup \{e_{j_0+1}, e_{j_0+3}, \ldots, e_{2p-2}\} \cup \Gamma \cup \{e\}$. We can see as before that $\tilde{F}_2$ induces a k-edge connected subgraph of $\tilde{G}$ and contains exactly $p-2$ edges from C. As $e \in \tilde{F}_2$, we obtain that $\gamma = p-2$, and therefore the lifting coefficient of e equals 2.

Finally, suppose that e is an edge of L_3 between two non consecutive node sets $[V_{i_0}, V_{j_0}]$ with $i_0, j_0 \in \{2, \ldots, 2p-1\}$, $j_0 > i_0 + 1$, and, say, i_0 is odd and j_0 is even (the proof is similar if i_0 and j_0 have the same parity). Here observe that any solution $\tilde{F} \subseteq \tilde{E}$ of the kECSP on $\tilde{G}$ must cover by edges from C the node sets $V_2, \ldots V_{i_0-1}$, $V_{i_0+1}, \ldots, V_{j_0-1}$ and $V_{j_0+1}, \ldots, V_{2p-1}$. By Lemma 2.3, $\tilde{F}$ must then use at least $\lceil \frac{i_0-2}{2} \rceil + \lceil \frac{j_0-i_0-1}{2} \rceil + \lceil \frac{2p-j_0-1}{2} \rceil = p-1$ edges from C. Thus $\gamma \geq p-1$. Moreover, as the edge set $\tilde{F}_3 = \{e_1, e_3, \ldots, e_{i_0-2}\} \cup \{e_{i_0+1}, e_{i_0+1}, \ldots, e_{2p-2}\} \cup \Gamma \cup \{e\}$ induces a k-edge connected subgraph of $\tilde{G}$ and contains exactly $p-1$ edges from C, we have that $\gamma = p-1$. Hence the lifting coefficient of e in $\tilde{G}$ is equal to 1.

Consequently the lifting coefficient of e equals 1 (respectively 2) (respectively 1) if $e \in L_1$ (respectively $e \in L_2$) (respectively $e \in L_3$). By Lemma 2.4, we then have that $a(e) \geq 1$ if $e \in L_1 \cup L_3$ and $a(e) \geq 2$ if $e \in L_2$, which ends the proof of the claim. ∎

In what follows, we are going to show that we also have $a(e) \leq 1$ (respectively $a(e) \leq 2$) (respectively $1 \leq a(e) \leq 2$) if $e \in L_1$ (respectively $e \in L_2$) (respectively $e \in L_3$). For this, let us consider a sequence $f_1, \ldots, f_t$, $t \geq 1$, of edges of L, and suppose that $f_1, \ldots, f_t$ are the edges that are added to G before e.

Suppose first that $e \in L_1$ and let us assume as before that $e \in [W_{j_0}, V_{i_0}]$ with $i_0 \in \{2, \ldots, 2p-1\}$ and even, and $j_0 \in \{1, 2\}$. Let $\hat{G} = (V, \hat{E})$ be the graph where $\hat{E} = E \cup \{f_1, \ldots, f_t, e\}$. Any solution $\hat{F} \subseteq \hat{E}$ of the kECSP on $\hat{G}$ must cover the node sets $V_2, \ldots, V_{i_0-1}$ and $V_{i_0+1}, \ldots, V_{2p-1}$ by edges from $(C \cup \{f_1, \ldots, f_t\}) \setminus L_4$. By Lemma 2.3, $\hat{F}$ must use at least $\lceil \frac{i_0-2}{2} \rceil + \lceil \frac{2p-i_0-1}{2} \rceil = p-1$ edges from $(C \cup \{f_1, \ldots, f_t\}) \setminus L_4$. As, by the claim above, $a(f) \geq 1$ for every edge $f \in (C \cup \{f_1, \ldots, f_t\}) \setminus L_4$, we have that $\gamma \geq p-1$ and hence by Lemma 2.4, we have that $a(e) \leq 1$. As, by the claim above $a(e) \geq 1$, this implies that $a(e) = 1$. Moreover, this holds independently on the order in which e is added to G.

Now consider an edge $e \in L_2$ and suppose that $e \in [V_{i_0}, V_{j_0}]$, with $i_0, j_0 \in \{2, \ldots, 2p-1\}$, $j_0 > i_0 + 1$, i_0 even and j_0 odd. Any solution $\hat{F} \subseteq \hat{E}$ of the kECSP on $\hat{G}$ must cover the node sets $V_2, \ldots, V_{i_0-1}, V_{i_0+1}, \ldots, V_{j_0-1}$ and $V_{j_0+1}, \ldots, V_{2p-1}$ by edges from $(C \cup \{f_1, \ldots, f_t\}) \setminus L_4$. By Lemma 2.3, $\hat{F}$ must use $\lceil \frac{i_0-2}{2} \rceil + \lceil \frac{j_0-i_0-1}{2} \rceil + \lceil \frac{2p-j_0-1}{2} \rceil = p-1$ edges of $(C \cup \{f_1, \ldots, f_t\}) \setminus L_4$. Thus $\gamma \geq p-2$ and therefore $a(e) \leq 2$. As, by the claim above, $a(e) \geq 2$, we get $a(e) = 2$.

If e is an edge of L_3, we show along the same line that $1 \leq a(e) \leq 2$.

In consequence, $a(e) = 1$ if $e \in L_1$, $a(e) = 2$ if $e \in L_2$, $1 \leq a(e) \leq 2$, which ends the proof of the theorem. ∎

APPENDIX B. PROOF OF THEOREM 2.5

First observe that, by conditions 1–5, G is $(k+1)$-edge connected and hence kECSP(G) is full dimensional. Let us denote inequality (2.19) by $ax \geq \alpha$ and let $\mathcal{F} = \{x \in k\text{ECSP}(G) \mid ax = \alpha\}$. Clearly, $\mathcal{F}$ is a proper face of kECSP(G). Now suppose that there exists a facet defining inequality $bx \geq \alpha$ such that $\mathcal{F} \subseteq \{x \in k\text{ECSP}(G) \mid bx = \alpha\}$. We will show that $b = a$.

Let e_i be an edge of $[V_i, V_{i+1}]$, $i = 1, \ldots, 2l+1$, and f_i and f'_i be edges of $[V_i, U_{i-1}]$ and $[V_i, U_i]$, respectively, for $i = 1, \ldots, 2l+1$. Let T_i be an edge subset of $[U_i, U_{i+1}]$ of $\frac{k-1}{2}$ edges, for $i = 1, \ldots, 2l+1$.

Let E_0 be the set of edges not in F and having both endnodes in the same element of π. First we will show that $b(e) = 0$ for all $e \in E_0 \cup F$. Let $i_0 \in \{1, \ldots, 2l+1\}$ and consider the edge sets

$$E_1 = \{e_{i_0+2r},\ r = 0, \ldots, l\} \cup \left\{f'_i,\ i = 1, \ldots, 2l+1\right\} \cup \left(\bigcup_{i=1}^{2l+1} T_i\right),$$

$$E_2 = E_1 \cup F \cup E_0.$$

Claim. *E_2 induces a k-edge connected subgraph of G.*

Proof. Let G_2 be the subgraph of G induced by E_2. As by condition 1, the graphs induced by the node sets W and V_i, U_i, $i = 1, \ldots, 2l+1$, are $(k+1)$-edge connected, it suffices to show that the graph obtained by contracting W and V_i, U_i, $i = 1, \ldots, 2l+1$, is k-edge connected. Let $\bar{G}_2 = (\bar{V}_2, \bar{E}_2)$ be that graph and $w, v_1, \ldots, v_{2l+1}, u_1, \ldots, u_{2l+1}$ the nodes of $\bar{G}_2$ where w corresponds to W, v_i to V_i and u_i to U_i, for $i = 1, \ldots, 2l+1$. Let $\delta(U)$ be a cut of $\bar{G}_2$ and let $\bar{G}'_2 = (\bar{V}'_2, \bar{E}'_2)$ the subgraph of $\bar{G}_2$ induced by $\{w, v_1, \ldots, v_{2l+1}\}$ and $\bar{G}''_2 = (\bar{V}''_2, \bar{E}''_2)$ the graph obtained from $\bar{G}_2$ by contracting $\{w, v_1, \ldots, v_{2l+1}\}$. Note that $\bar{E}'_2 \cap \bar{E}''_2 = \emptyset$ and $\bar{E}_2 = \bar{E}'_2 \cup \bar{E}''_2$. Also note that $\bar{G}'_2$ is $(k-1)$-edge connected and that $\bar{G}''_2$ is a k-edge connected wheel. Thus if U does not intersect $\{w, v_1, \ldots, v_{2l+1}\}$, then $\delta(U)$ is a cut of $\bar{G}''_2$ and hence $|\delta(U)| \geq k$. If U intersects $\{w, v_1, \ldots, v_{2l+1}\}$, then $\delta(U)$ contains at least $k-1$ edges from $\bar{E}'_2$. However, in this case $\delta(U)$ also contains at least one edge from $\bar{E}''_2$. Thus we have that $|\delta(U)| \geq k$, and the statement follows.

Note that there are $k+1$ edges incident to V_{i_0} in the graph induced by E_2. Now, observe that for any edge $e \in F_{i_0}$, one can show in a similar way as in the claim above that $E'_2 = E_2 \setminus \{e\}$ also induces a k-edge connected subgraph of G. As x^{E_2} and $x^{E'_2}$ belong to $\mathcal{F}$, it follows that $bx^{E_2} = bx^{E'_2} = \alpha$, implying that $b(e) = 0$ for all $e \in F_{i_0}$. As i_0 is arbitrarily chosen, we obtain that $b(e) = 0$ for all $e \in F$. Moreover, as the subgraphs induced by $W, V_1, \ldots, V_{2l+1}, U_1, \ldots, U_{2l+1}$ are all $(k+1)$-edge connected, the subgraph induced by $E_2 \setminus \{e\}$, for all $e \in E_0$, is also k-edge connected. This yields as before $b(e) = 0$ for all $e \in E_0$. Thus $b(e) = 0$ for all $e \in F \cup E_0$.

Next, we will show that $b(e) = a(e)$ for all $e \in \delta(\pi) \setminus F$. Let g_i be a fixed edge of T_i and let $T'_i = T_i \setminus \{g_i\}$, for $i = 1, \ldots, 2l+1$. Consider the edge sets

$$E_3 = \left\{f_i, f'_i, i = 1, \ldots, 2l+1\right\} \cup \left(\bigcup_{i=1}^{l} T_{2i}\right) \cup T_{2l+1} \cup \left(\bigcup_{i=0}^{l-1} T'_{2i+1}\right),$$

$$E_4 = E_3 \cup F \cup E_0,$$
$$E'_4 = (E_4 \setminus g_{2l+1}) \cup \{g_1\}.$$

Note that $g_1 \notin T'_1$ and thus $g_1 \notin E_4$, and that $g_{2l+1} \in E_4$. The edge sets E_4 and E'_4 can be obtained from E_2 using recursively the edge-swapping operation of Lemma 2.2. Hence both E_4 and E'_4 induce k-edge connected subgraphs of G. Moreover, we have that x^{E_4} and $x^{E'_4}$ belong to $\mathcal{F}$. Thus $bx^{E_4} = bx^{E'_4} = \alpha$ and therefore $b(g_{2l+1}) = b(g_1)$. As g_1 and g_{2l+1} are arbitrary edges of T_1 and T_{2l+1}, respectively, it follows that $b(e) = b(e')$ for all $e \in T_1$ and $e' \in T_{2l+1}$. Moreover, we have that T_1 and T_{2l+1} are arbitrary subsets of $[U_1, U_2]$ and $[U_{2l+1}, U_1]$, respectively. This implies that $b(e) = b(e')$ for all $e \in [U_1, U_2]$ and $e' \in [U_{2l+1}, U_1]$. Consequently, by symmetry, we get

$$b(e) = \rho \text{ for all } e \in [U_i, U_{i+1}], i = 1, \ldots, 2l+1, \quad \text{(B.1)}$$

for some $\rho \in \mathbb{R}$.

Now let

$$E_5 = (E_4 \setminus \{f_1\}) \cup \{e_{2l+1}\}.$$

Using Lemma 2.2 and the fact that E_4 induces a k-edge connected subgraph of G, we have that E_5 induces a k-edge connected subgraph of G. Moreover, x^{E_5} belongs to $\mathcal{F}$, implying that $bx^{E_4} = bx^{E_5} = \alpha$. Hence $b(f_1) = b(e_{2l+1})$. In a similar way, we can show that $b(f'_{2l+1}) = b(e_{2l+1})$. As f_1, f'_{2l+1} and e_{2l+1} are arbitrary edges of $[U_{2l+1}, V_1]$, $[V_{2l+1}, U_{2l+1}]$ and $[V_{2l+1}, V_1]$, respectively, we obtain that $b(e)$ is the same for all $e \in [U_{2l+1}, V_1] \cup [V_{2l+1}, U_{2l+1}] \cup [V_{2l+1}, V_1]$. By exchanging the roles of V_{2l+1}, V_1, U_{2l+1} and V_i, V_{i+1}, U_i, for $i = 1, \ldots, 2l$, we obtain by symmetry that

$$b(e) = \rho'_i \text{ for all } e \in [U_i, V_i] \cup [V_i, V_{i+1}] \cup [V_{i+1}, U_i], \quad \text{(B.2)}$$

$i = 1, \ldots, 2l+1$, for some $\rho'_i \in \mathbb{R}$.

Consider the edge set

$$E'_5 = (E_4 \setminus \{f_1\}) \cup \{e_1\}.$$

Similarly, we can show that E'_5 induces a k-edge connected subgraph of G. As x^{E_4} and $x^{E'_5}$ belong to $\mathcal{F}$, it follows in a similar way that $b(e_1) = b(f_1)$. From (B.2), we have that $\rho'_1 = \rho'_{2l+1}$. By symmetry, it then follows that $\rho'_i = \rho'_j$ for $i, j = 1, \ldots, 2l+1$, $i \neq j$, and therefore

$$b(e) = \rho' \text{ for all } e \in [U_i, V_i] \cup [V_i, V_{i+1}] \cup [V_{i+1}, U_i], \quad \text{(B.3)}$$

for $i = 1, \ldots, 2l+1$, for some $\rho' \in \mathbb{R}$.

Let $e \in ([V_{2l+1}, W] \setminus F_{2l+1}) \cup [V_{2l+1}, V_j], j \in \{2, \ldots, 2l-1\}$. As before, we can observe that $E_6 = (E_4 \setminus \{f'_{2l+1}\}) \cup \{e\}$ induces a k-edge connected subgraph of G. As $x^{E_6} \in \mathcal{F}$, this implies that $bx^{E_6} = bx^{E_4} = \alpha$ and hence $b(e) = b(f'_{2l+1})$. By (B.3), we then obtain that $b(e) = \rho'$ for all $e \in ([V_{2l+1}, W] \setminus F_{2l+1}) \cup [V_{2l+1}, V_i]$ for $i \in \{2, \ldots, 2l-1\}$. By exchanging the roles of V_{2l+1} and V_i, $i = 1, \ldots, 2l$, we obtain by symmetry that $b(e) = \rho'$ for all $e \in ([V_i, W] \setminus F_i) \cup [V_i, V_j]$, $i = 1, \ldots, 2l+1$ and $j \in \{1, \ldots, 2l+1\} \setminus \{i-1, i, i+1\}$.

For any edge e between U_{2l+1} and either W, U_j, $j \in \{1, \ldots, 2l+1\} \setminus \{1, 2l, 2l+1\}$, or V_t, $t \in \{1, \ldots, 2l+1\} \setminus \{1, 2l+1\}$, we can show, using Lemma 2.2 and the fact that E_4 induces a k-edge connected subgraph of G, that

$$E_7 = \left(E_4 \setminus \left\{f'_{2l+1}, f_1\right\}\right) \cup \{e, e_{2l+1}\}$$

also induces a k-edge connected subgraph of G. As x^{E_4} and x^{E_7} belong to $\mathcal{F}$, we have that $bx^{E_7} = bx^{E_4} = \alpha$ and $b(f'_{2l+1}) + b(f_1) = b(e) + b(e_{2l+1})$. As by (B.3), $b(f'_{2l+1}) = b(f_1) = b(e_{2l+1}) = \rho'$, we get $b(e) = \rho'$. Here again, by exchanging the roles of U_{2l+1} and U_i, $i = 1, \ldots, 2l$, we obtain that $b(e) = \rho'$ for all $e \in [U_i, W] \cup [U_i, U_j] \cup [U_i, V_t]$, $i = 1, \ldots, 2l+1$, $j \in \{1, \ldots, 2l+1\} \setminus \{i, i+1\}$ and $t \in \{1, \ldots, 2l+1\} \setminus \{i-1, i, i+1\}$.

As x^{E_2} and x^{E_4} belong to $\mathcal{F}$, we have that $bx^{E_2} = bx^{E_4} = \alpha$. Thus from (B.1) and (B.3), we obtain that $\rho = \rho'$, and in consequence, the edges of $E \setminus (E_0 \cup F)$ have all the same coefficient in $bx \geq \alpha$. Since $ax^{E_2} = bx^{E_2} = \alpha$, this yields $b(e) = 1$ for all $e \in E \setminus (E_0 \cup F)$.

Thus we obtain that $b = a$, which ends the proof of the theorem. ∎

APPENDIX C. PROOF OF THEOREM 2.8

Note that because G_π is outerplanar and conditions 1 and 2 hold, G is $(k+1)$-edge connected. It then follows that kECSP(G) is full dimensional. Let us denote by $ax \geq \alpha$ the SP-partition inequality induced by π and let $\mathcal{F} = \{x \in k\text{ECSP}(G) | ax = \alpha\}$. Clearly, $\mathcal{F}$ is a proper face of kECSP(G). Now suppose that there exists a facet defining inequality $bx \geq \alpha$ different from the trivial inequalities such that $\mathcal{F} \subseteq \{x \in k\text{ECSP}(G) | bx = \alpha\}$. We will show as before that $b = a$.

Let T_i be an edge subset of $[V_i, V_{i+1}]$, $i = 1, \ldots, p$, of $\frac{k+1}{2}$ edges and let $T'_i = T_i \setminus \{g_i\}$, where g_i is a fixed edge of T_i. Consider

$$E_0 = \bigcup_{i=1}^{p} E(V_i),$$
$$E_1 = \left(\bigcup_{i=1}^{p} T_i\right) \setminus \{g_{i_0}\} \text{ for some } i_0 \in \{1, \ldots, p\},$$
$$E_2 = E_1 \cup E_0.$$

Note that $g_{i_0} \notin E_2$ and $g_{i_0+1} \in E_2$. As by condition 1, the subgraphs induced by the node sets $V_1, \ldots, V_p$ are $(k+1)$-edge connected, it is not hard to see that E_2 and $E'_2 = (E_2 \setminus \{g_{i_0+1}\}) \cup \{g_{i_0}\}$ induce k-edge connected subgraphs of G. As x^{E_2} and $x^{E'_2}$ belong to $\mathcal{F}$, we have that $bx^{E_2} = bx^{E'_2} = \alpha$ and hence $b(g_{i_0}) = b(g_{i_0+1})$. As g_{i_0} and g_{i_0+1} are arbitrary edges of T_{i_0} and T_{i_0+1}, respectively, it follows that $b(e) = b(e')$ for all $e \in T_{i_0}$ and $e' \in T_{i_0+1}$. Moreover, as T_{i_0} and T_{i_0+1} are arbitrary subsets of $[V_{i_0}, V_{i_0+1}]$

and $[V_{i_0+1}, V_{i_0+2}]$, respectively, we obtain that $b(e) = b(e')$ for all $e \in [V_{i_0}, V_{i_0+1}]$ and $e' \in [V_{i_0+1}, V_{i_0+2}]$, $i_0 = 1, \ldots, p$. Consequently, by symmetry, we get

$$b(e) = b(e') \text{ for all } e, e' \in \bigcup_{i=1}^{p} [V_i, V_{i+1}]. \tag{C.1}$$

Now let $e \in [V_{i_0}, V_{j_0}]$, $i_0, j_0 \in \{1, \ldots, p\}$ with $|i_0 - j_0| > 1$. Note that $T_0 = T_p$, $T_{-1} = T_{p-1}$ and $T'_0 = T'_p$. Consider the edge sets

$$E_4 = (E_2 \setminus \{g_{i_0-1}\}) \cup \{e\},$$
$$E'_4 = (E_4 \setminus \{e\}) \cup \{g_{i_0}\}.$$

Using Lemma 2.2 and the fact that E_2 induces a k-edge connected subgraph of G, we can see that E_4 and E'_4 induce k-edge connected subgraphs of G. As x^{E_4} and $x^{E'_4}$ belong to $\mathcal{F}$, it follows that $bx^4 = bx^{E'_4} = \alpha$, and hence $b(e) = b(g_{i_0})$. By (C.1) this yields

$$b(e) = b(e') \text{ for all } e, e' \in \delta(\pi).$$

As $ax^{E_2} = bx^{E_2} = \alpha$, we obtain that $b(e) = 1$ for all $e \in \delta(\pi)$.

Next, we will show that $b(e) = 0$ for all $e \in E_0$. Consider the edge set

$$E_5 = E_2 \setminus \{e\} \text{ for some } e \in E_0.$$

As $G[V_i]$, $i = 1, \ldots, p$, are $(k+1)$-edge connected, E_5 induces a k-edge connected subgraph of G. As x^{E_2} and x^{E_5} belong to $\mathcal{F}$, we have that $bx^{E_2} = bx^{E_5} = \alpha$, and thus $b(e) = 0$ for all $e \in E_0$.

In consequence we get $b = a$ and the proof is complete. ∎

REFERENCES

[1] ABACUS, A branch-and cut system, Available at: http://www.informatik.uni-koeln.de/abacus.

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

[3] M. Baïou and A.R. Mahjoub, Steiner 2-edge connected subgraph polytopes on series-parallel graphs, SIAM J Discrete Math 10 (1997), 505–514.

[4] F. Barahona and A.R. Mahjoub, On two-connected subgraph polytopes, Discrete Math 147 (1995), 19–34.

[5] D. Bienstock, E.F. Brickel, and C.L. Monma, On the structure of minimum weight k-connected networks, SIAM J Discrete Math 3 (1990), 320–329.

[6] S.C. Boyd and T. Hao, An integer polytope related to the design of survivable communication network, SIAM J Discrete Math 6 (1993), 612–630.

[7] S. Chopra, The k-edge connected spanning subgraph polyhedron, SIAM J Discrete Math 7 (1994), 245–259.

[8] G. Cornuéjols, J. Fonlupt, and D. Naddef, The traveling salesman problem on a graph and some related polyhedra, Math Prog 33 (1985), 1–27.

[9] R. Coullard, A. Rais, R.L. Radin, and D.K. Wagner, Linear time algorithm for the 2-connected Steiner subgraph problem on special classes of graphs, Networks 23 (1993), 195–206.

[10] R. Coullard, A. Rais, R.L. Radin, and D.K. Wagner, The Dominant of the 2-connected Steiner subgraph polytope for W_4-free graphs, Discrete Appl Math 66 (1996), 33–43.

[11] Cplex, Available at: http://www.ilog.com.

[12] M. Didi Biha, Graphes k-arêtes connexes et polyèdres, Thèse de doctorat, Université de Bretagne Occidentale, Brest, 1998.

[13] I. Diarrassouba and L. Slama, Les inégalités de SP-partition pour le problème du sous-graphe k-arête connexe, Research Report, LIMOS RR-07-13, 2007.

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

[15] M. Didi Biha and A.R. Mahjoub, Steiner k-edge connected subgraph polyhedra, J Combinat Optim 4 (2000), 131–134.

[16] M. Didi Biha and A.R. Mahjoub, The k-edge connected subgraph problem I. Polytopes and critical extreme points, Linear Algebra Appl 381 (2004), 117–139.

[17] M. Didi Biha, A.R. Mahjoub, and L. Slama, On the separation of partition inequalities, Proc INOC 2005, Lisbon, Pontugal, 2005, pp. 500–505.

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

[19] J. Fonlupt and D. Naddef, The traveling salesman problem in graphs with some excluded minors, Math Prog 53 (1992), 147–172.

[20] G.N. Frederickson and J. Jàjà, On the relationship between the biconnectivity augmentations and traveling salesman problem, Theor Computer Sci 13 (1982), 189–201.

[21] M.R. Garey and D.J. Johnson, Computer and intractability: A guide to the theory of NP-completness, Freeman, San Francisco, 1979.

[22] A. Goldberg and R.E. Tarjan, A new approach to the maximum flow problem, J Assoc Comput Machinery 35 (1988), 921–940.

[23] R.E. Gomory and T.C. Hu, Multi-terminal network flows, J Soc Ind Appl Math 9 (1961), 551–570.

[24] M. Grötschel and C.L. Monma, Integer polyhedra arising from certain network design problems with connectivity constraints, SIAM J Discrete Math 3 (1990), 502–523.

[25] M. Grötschel, C.L. Monma, and M. Stoer, Polyhedral approches to network survivability, Reliability of computer and Communication networks, Vol 5, Series Discrete Mathematics and Computer Science, F. Roberts, F. Hwang, and C.L. Monma (editors), AMS/ACM, 1991, pp. 121–141.

[26] M. Grötschel, C.L. Monma, and M. Stoer, Polyhedral and computational investigations arising for designing communication networks with high survivability requirements, Oper Res 43 (1995), 1012–1024.

[27] D. Gusfield, Very simple method for all pairs network flow analysis, Soc Ind Appl Math 009 (1990), 143–155.

[28] J. Hao and J.B. Orlin, A faster algorithm for finding the minimum cut in a directed graph, J Algorithms 17 (1992), 424–446.

[29] H. Kerivin and A.R. Mahjoub, Design of survivable networks: A survey, Networks 46 (2005), 1–21.

[30] H. Kerivin, A.R. Mahjoub, and C. Nocq, (1,2)-survivable networks: Facets and branch-and-cut, The sharpest cut, MPS-SIAM Series in Optimization, M. Grötschel (Editor), 2004, pp. 121–152.

[31] C.-W. Ko and C.L. Monma, Heuristics for designing highly survivable communication networks, 1989, New Jersey.

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

[33] A.R. Mahjoub, On perfectly two-edge connected subgraphs and polyhedra, Discrete Math 170 (1997), 153–172.

[34] C.L. Monma, B.S. Munson, and W.R. Pulleyblank, Minimum weight two-connected spanning networks, Oper Res 37 (1989), 153–171.

[35] G.L. Nemhauser and L.A. Wolsey, Integer and combinatorial optimization, Wiley Editions, 1988, pp. 259–295.

[36] S. Thienel, ABACUS—A branch-and-cut system, PhD Thesis, Universität zu Köln, 1995.

[37] TSPLIB, Available at: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.