Computers & Industrial Engineering 112 (2017) 690–705

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

The survivable k-node-connected network design problem: Valid inequalities and Branch-and-Cut

Meriem Mahjoub [a,*], I. Diarrassouba [b], A.R. Mahjoub [a], R. Taktak [c]

[a] Université Paris-Dauphine, PSL, CNRS UMR 7243 LAMSADE, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France
[b] Laboratoire LMAH, Université du Havre, 25 Rue Philippe Lebon, 76600 Le Havre, France
[c] ISIMS, Pôle Technologique de Sfax, Route de Tunis Km 10, 3021 Sfax, Tunisia

ARTICLE INFO

Article history:
Available online 2 March 2017

Keywords:
k-node-connected graph
Polytope
Facets
Separation
Branch-and-Cut

ABSTRACT

In this paper we consider the k-node-connected subgraph problem. We propose an integer linear programming formulation for the problem and investigate the associated polytope. We introduce further classes of valid inequalities and discuss their facial aspect. We also devise separation routines, investigate the structural properties of the linear relaxation and discuss some reduction operations that can be used in a preprocessing phase for the separation. Using these results, we devise a Branch-and-Cut algorithm and present some computational results.

1. Introduction

The design of survivable networks is an important issue in telecommunications. The aim is to conceive cheap, efficient and reliable networks with specific characteristics and requirements on the topology. Survivability is generally expressed in terms of connectivity in the network. The level of connectivity depends on the need of each telecommunication operator. We may have to conceive several paths to link each pair of nodes to ensure the transmission in case of disconnection or breakdown, all this at the cheapest possible cost. As we can see in Grötschel, Monma, and Stoer (1995) and Kerivin and Mahjoub (2015), the most frequent and useful case in practice is the uniform topology. This means that the nodes of the network have all the same importance and it is required that between every pair of nodes there are at least k edge- (node-) disjoint paths, where k is a fixed integer such that $k \geqslant 2$. Thus the network will be still functional when at most $k-1$ edges fail. The underlaying problem is to determine, given weights on the possible links of the network, a minimum weight network satisfying the edge or the node connectivity. This paper deals with the node connectivity of the problem.

A graph $G=(V,E)$ is called *k-node* (resp. *k-edge*) *connected* ($k \geqslant 0$) if for every pair of nodes $i,j \in V$, there are at least k node-disjoint (resp. edge-disjoint) paths between i and j. Given a graph $G=(V,E)$ and a weight function c on E that associates with an edge $e \in E$ a weight $c(e) \in \mathbb{R}$, the *k-node-connected subgraph problem* (kNCSP for short) is to find a k-node-connected spanning subgraph $H=(V,F)$ of G such that $\sum_{e \in F} c(e)$ is minimum. The kNCSP has applications in communication and transportation networks (Bendali, Diarrassouba, Didi Biha, Mahjoub, and Mailfert (2010); Grötschel and Monma (1990); Grötschel, Monma, and Stoer (1991, 1992, 1995)). The kNCSP is NP-hard for $k \geqslant 2$ (Garey & Johnson, 1979). The edge version of the problem has been widely studied in the literature (Bendali et al. (2010); Chopra (1994); Didi Biha and Mahjoub (1996); Grötschel and Monma (1990); Grötschel et al. (1991, 1992, 1995); Mahjoub (1994)). However, the kNCSP has been particulary considered for $k=2$ (see Diarrassouba, Kutucu, and Mahjoub (2016) and Mahjoub and Nocq (1999)). A little attention has been given for the high connectivity case where $k \geqslant 3$. The kNCSP has been studied by Grötschel et al. (1991, 1990, 1992, 1995) within a more general survivability model. Grötschel et al. study the model from a polyhedral point of vue, and propose cutting plane algorithms (Grötschel and Monma (1990); Grötschel et al. (1992, 1995)).

Diarrassouba et al. (2016) consider the 2NCSP with bounded lengths. Here it is supposed that each path does not exceed L edges for a fixed integer $L \geqslant 1$. They investigate the structure of the associated polytope when $L \leqslant 3$ and propose a Branch-and-Cut algorithm. Mahjoub and Nocq (1999) discuss the linear relaxation of the 2NCSP(G). They describe some structral properties and characterize which they called extreme points of rank 1.

In this article, we consider the kNCSP from a polyhedral point of view. We introduce further classes of valid inequalities for the associated polytope, discuss their facial aspect and devise a Branch-and-Cut algorithm.

* Corresponding author.
E-mail address: meriem.mahjoub@dauphine.fr (M. Mahjoub).

http://dx.doi.org/10.1016/j.cie.2017.03.007
0360-8352/

The paper is organised as follows. In the following section, we give an integer programming formulation for the problem. In Section 3 we investigate the kNCSP polytope and present several classes of valid inequalities. Then, in Section 4, we discuss the conditions under which these inequalities define facets of the polytope. In Sections 5 and 6, we consider the polytope associated with the linear relaxation of the problem and present some structural properties as well as some reduction operations. Section 7 is devoted to the Branch-and-Cut algorithm we have developed for the problem and in Section 8 we give some experimental results. Finally, in Section 9, we give some concluding remarks.

In the rest of this section, we give some notations. We will denote a graph by $G = (V, E)$ where V is the *node set* and E is the *edge set*. Given $F \subseteq E, c(F)$ will denote $\sum_{e\in F} c(e)$. For $W \subseteq V$, we let $\overline{W} = V \setminus W$. If $W \subset V$ is a node subset of G, then $\delta_G(W)$ will denote the set of edges in G having one node in W and the other in $\overline{W}$. We will write $\delta(W)$ if the meaning is clear from the context. For $W \subset V$, we denote by $E(W)$ the set of edges of G having both endnodes in W and by $G[W]$ the subgraph induced by W. Given node subsets $W_1, \dots, W_p \subset V, p \geqslant 2$, we denote by $\delta_G(W_1, \dots, W_p)$ the set of edges of G between the sets $W_1, \dots, W_p$. For $U \subset V, W \subset V$ and $U \cap W = \emptyset$, the edge set $[U, W]$ will denote the set of edges between U and W.

A *matching* of G is a set of pairwise nonadjacent edges.

Let F be an edge subset of E, then the *incidence vector* of F, denoted by x^F, is the $0-1$ vector defined by

$$x^F(e) = \begin{cases} 1 & \text{if } e \in F \\ 0 & \text{otherwise.} \end{cases}$$

2. Formulation

Let $F \subseteq E$ be an edge subset of G. Then F induces a solution of the kNCSP for G, that is, the subgraph of G induced by F is k-node-connected, if x^F satisfies the following inequalities

$$x(e) \geqslant 0, \; e \in E, \tag{1}$$
$$x(e) \leqslant 1, \; e \in E, \tag{2}$$
$$x(\delta_G(W)) \geqslant k, \text{ for all } W \subsetneq V \text{ with } W \neq \emptyset, \tag{3}$$
$$x(\delta_{G\setminus Z}(W)) \geqslant k - |Z|, \text{ for all } Z \subseteq V \text{ such that } 1 \leqslant |Z| \leqslant k-1, \text{ and all } W \subsetneq V \setminus Z \text{ with } W \neq \emptyset. \tag{4}$$

Conversely, any integer solution of the system above is the incidence vector of the edge set of a k-node-connected subgraph of G. Hence, the kNCSP is equivalent to

$$\min\{cx | x \text{ satisfies } (1)–(4) \text{ and } x \in \mathbb{Z}^E_+\}. \tag{5}$$

Constraints (3) and (4) are called *cut* and *node-cut* inequalities, respectively. The convex hull of all integer solutions of (1)–(4), denoted by kNCSP(G), will be called kNCSP(G) *the k-node-connected subgraph problem polytope*.

We will also denote by $P(G, k)$ the polytope described by constraints (1)–(4).

In what follows we give an alternative formulation for the problem. This consists in restricting the node-cut inequalities (4) to the node sets $Z \subset V$ such that $|Z| = k - 1$. We hence consider the following set of inequalities

$$x(\delta_{G\setminus Z}(W)) \geqslant 1, \emptyset \neq Z \subseteq V, |Z| = k-1, \; \emptyset \neq W \subset V \setminus Z. \tag{6}$$

Theorem 1. *The kNCSP is equivalent to*

$$\min\{cx | x \text{ satisfies } (1)–(3), (6) \text{ and } x \in \mathbb{Z}^E_+\}. \tag{7}$$

Proof. It suffices to show that any integer solution $\bar{x}$ of (1)–(3), (6) also satisfies (4). For this we will show that if $\bar{x}$ satisfies all inequalities $x(\delta_{G\setminus Z}(W)) \geqslant k - |Z|$ with $|Z| = t$ for some $t \in \{k-1, \dots, 2\}$, then $\bar{x}$ satisfies $x(\delta_{G\setminus Z'}(W')) \geqslant k - |Z'|$ for all $Z' \subset V$ with $|Z'| = t - 1$ and $W' \subset V \setminus Z'$. Indeed, first note that either $|V \setminus (W' \cup Z')| \geqslant 2$ or $|W'| \geqslant 2$ or both. In fact, if $|V \setminus (W' \cup Z')| = |W'| = 1$, then $|Z'| = n - 2 (= t - 1)$. But this implies that $t = n - 1$, and, as $t \leqslant k - 1$, it follows that $k \geqslant n$, which is impossible. In what follows we suppose, w.l.o.g., that $|V \setminus (W' \cup Z')| \geqslant 2$. We claim that there is at least one node, say u, in $V \setminus (W' \cup Z')$ such that $\bar{x}([u, W']) \geqslant 1$. In fact, let $u_0 \in V \setminus (W' \cup Z')$. Let $Z = Z' \cup \{u_0\}$. By our assumption, $\bar{x}(\delta_{G\setminus Z}(W')) \geqslant k - |Z| = k - t$. As $t \leqslant k - 1$, it follows that $\bar{x}(\delta_{G\setminus Z}(W')) \geqslant 1$. Therefore there is a node u in $V \setminus (W' \cup Z')$ such that $\bar{x}([u, W']) \geqslant 1$.

Now let $u \in V \setminus (W' \cup Z')$ $\bar{x}([u, W']) \geqslant 1$, $Z^* = Z' \cup \{u\}$. We have $|Z^*| = t$. Again, by our assumption, we have that $\bar{x}(\delta_{G\setminus Z^*}(W')) = \bar{x}(\delta_{G\setminus Z'}(W')) - \bar{x}([u, W']) \geqslant k - |Z^*| = k - t$. As $\bar{x}([u, W']) \geqslant 1$, it then follows that $\bar{x}(\delta_{G\setminus Z'}(W')) = \bar{x}(\delta_{G\setminus Z^*}(W')) + \bar{x}([u, W']) \geqslant k - t + 1 = k - |Z'|$. □

As before, we will denote by $Q(G, k)$ the polytope associated with the linear relaxation of (7). Clearly, $P(G, k) \subseteq Q(G, k)$. Moreover, the two polytopes may be different, that is $P(G, k) \neq Q(G, k)$, for some graph G and connectivity k. For example, consider the graph and the solution of Fig. 1 for $k = 3$. The solution satisfies the cut inequalities and the node-cut inequalities with $|Z| = k - 1 = 2$, and violates a node-cut inequality with $|Z| = k - 2 = 1$. Indeed, for $Z = \{v_7\}$ and $W = \{v_1, v_2, v_3\}$, $x(\delta_{G\setminus Z}(W)) < 2$. Thus, formulation (5) may produce a better linear relaxation than (7). We will hence consider formulation (5) for solving the kNCSP.

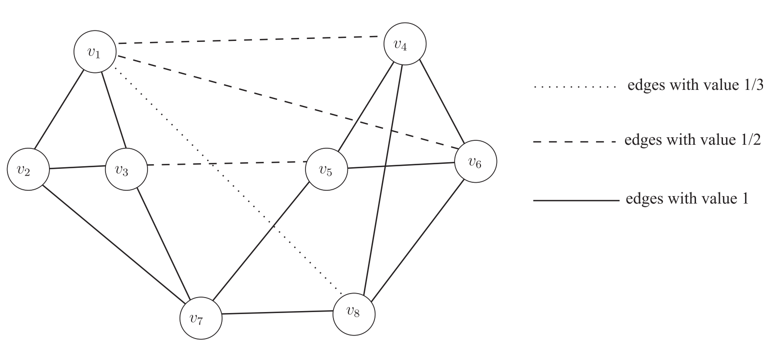

Fig. 1. A solution of $Q(G, k) \setminus P(G, k)$ for $k = 3$.

In the next sections, we investigate the polytope kNCSP(G) and describe some valid inequalities.

3. Dimension and valid inequalities

In this section, we will discuss the polytope kNCSP(G). We will establish its dimension and describe some classes of valid inequalities.

3.1. Dimension

Let $G = (V, E)$ be a graph. An edge e is said to be *essential* if the solutions set of kNCSP($G \setminus e$) is empty. Let E^* be the set of essential edges of kNCSP. We have the following result.

Theorem 2. *dim(kNCSP(G))=* $|E| - |E^*|$.

Proof. Let $e \in E^*$. Then, $x(e) = 1$ for every solution x of kNCSP(G). Then $dim(k\text{NCSP}(G)) \leqslant |E| - |E^*|$. Now, observe that the edge sets $S_e = E \setminus \{e\}, e \in E \setminus E^*$, and E form $|E| - |E^*| + 1$ solutions of the kNCSP. Moreover, the incidence vectors of these solutions are affinely independent. Therefore, $dim(k\text{NCSP}(G)) \geqslant |E| - |E^*|$. Thus, the result follows. □

Corollary 1. *kNCSP(G) is full-dimensional if and only if G is* $(k+1)$*-node-connected.*

Now we describe some classes of valid inequalities for kNCSP (G). One can easily see that any solution of the kNCSP on G is also solution of the kECSP on G. Thus, any valid inequality for the kECSP polytope on G is also valid for kNCSP(G).

In the following, we introduce a notation that will be used throughout the remainder of the paper. Given a partition $\pi = (V_1, \ldots, V_p), p \geqslant 2$, we will denote by G_π the subgraph induced by π, that is, the graph obtained by contracting the sets $V_i, i = 1, \ldots, p$. We will denote by $\delta_G(V_1, \ldots, V_p)$ the set of edges of G_π, that is, the edges that have their endnodes in different elements of π.

3.2. Node-partition inequalities

Grötschel et al. (1991) introduce a class of valid inequalities for a more general version of the kNCSP as follows. Consider a subset $Z \subset V$, such that $|Z| \leqslant k-1$, and let $V_1, \ldots, V_p, p \geqslant 2$ be a partition of $V \setminus Z$. Then the inequality

$$x(\delta_{G\setminus Z}(V_1, \ldots, V_p)) \geqslant \begin{cases} \left\lceil \frac{p(k-|Z|)}{2} \right\rceil & \text{if} \quad |Z| \leqslant k-2, \\ p-1 & \text{if} \quad |Z| = k-1, \end{cases} \tag{8}$$

is valid for the kNCSP(G). Inequalities of type (8) are called *node-partition inequalities*.

3.3. SP-node-partition inequalities

Now we introduce a class of inequalities called SP-node-partition inequalities, which generalize the so-called SP-partition inequalities introduced by Didi Biha and Mahjoub (1996) for the kECSP(G), where kECSP(G) is the convex hull of the solutions of the k-edge-connected subgraph problem. These latter inequalities are defined as follows. Let $\pi = (V_1, \ldots, V_p)$ be a partition of V such that the graph G_π is series-parallel. Recall that a graph is series-parallel if it is not contractible to K_4, the complete graph on four nodes. The SP-partition inequality associated with π is given by

$$x(\delta_G(V_1, \ldots, V_p)) \geqslant \left\lceil \frac{k}{2} \right\rceil p - 1. \tag{9}$$

Didi Biha and Mahjoub (1996) showed that these inequalities are valid for the kECSP(G), for every $k \geqslant 1$.

For the kNCSP, we introduce a similar type of inequalities. Let $Z \subset V$ such that $|Z| \leqslant k-1$ and $k - |Z|$ is odd, and consider a partition $\pi = (V_1, \ldots, V_p)$ of $V \setminus Z$ such that $(G \setminus Z)_\pi$ is series-parallel. The *SP-node-partition inequality* associated with π is

$$x(\delta_{G\setminus Z}(\pi)) \geqslant \left\lceil \frac{k-|Z|}{2} \right\rceil p - 1. \tag{10}$$

Theorem 3. *The SP-node-partition inequalities (10) are valid for kNCSP(G).*

Proof. Let $x \in k\text{NCSP}(G)$ and consider x' the restriction of x on $G \setminus Z$. As $x' \in (k-|Z|)\text{ECSP}(G \setminus Z)$, and the SP-partition inequalities (9) are valid for $(k-|Z|)\text{ECSP}(G \setminus Z)$, we have

$$x(\delta_{G\setminus Z}(V_1, \ldots, V_p)) = x'(\delta_{G\setminus Z}(V_1, \ldots, V_p)) \geqslant \left\lceil \frac{k-|Z|}{2} \right\rceil p - 1,$$

which proves the result. □

Chopra (1994) (see also Didi Biha and Mahjoub (1996)) described a lifting procedure for inequalities (10). This can be easily extended to the SP-node-partition inequalities. Let $G = (V, E)$ be a graph and $k \geqslant 3$ an odd integer. Consider the graph $G' = (V, E \cup T)$ obtained from G by adding an edge set T. Let $Z \subset V$ and $\pi = (V_1, \ldots, V_p)$ be a partition of $V \setminus Z$ such that $(G \setminus Z)_\pi$ is series-parallel. Then the *lifted SP-node-partition inequality* induced by π is

$$x(\delta_{G\setminus Z}(V_1, \ldots, V_p)) + \sum_{e \in T \cap \delta_{G'}(V_1, \ldots, V_p)} a(e)x(e) \geqslant \left\lceil \frac{k-|Z|}{2} \right\rceil p - 1, \tag{11}$$

where $a(e)$ is the length, that is to say the number of edges, of the shortest path in $(G \setminus Z)_\pi$ between the endnodes of e, for all $e \in T \cap \delta_{G'}(V_1, \ldots, V_p)$.

Chopra (1994) shows that inequalities (11) are valid for kECSP (G). Therefore they are also valid for kNCSP(G).

3.4. F-node-partition inequalities

Let $G = (V, E)$ be a graph and Z a node subset of V. Let $\pi = (V_0, V_1, \ldots, V_p)$ be a partition of $V \setminus Z$ and F an edge subset of $\delta_{G\setminus Z}(V_0)$. Let $Z_i \subset Z$ be the set of nodes of Z adjacent to the nodes in $V_i, i = 1, \ldots, p$. Suppose that $|Z_i| \leqslant k-1$, for $i = 1, \ldots, p$, and $Z = \bigcup_{i=1,\ldots,p} Z_i$. The following inequality

$$x(\delta_{G\setminus Z}(\pi) \setminus F) \geqslant \left\lceil \frac{\sum_{i=1}^{p}(k-|Z_i|) - |F|}{2} \right\rceil \tag{12}$$

is called an *F-node-partition inequality*.

Theorem 4. *F-node partition inequalities are valid for the kNCSP(G).*

Proof. Consider the following valid inequalities

$$x(\delta_{G\setminus Z_i}(V_i)) \geqslant k - |Z_i|, \quad \text{for all } i = 1, \ldots, p,$$
$$-x(e) \geqslant -1, \quad \text{for all } e \in F,$$
$$x(e) \geqslant 0, \quad \text{for all } e \in \delta_{G\setminus Z}(V_0) \setminus F.$$

By summing these inequalities, we obtain

$$2x(\delta_{G\setminus Z}(\pi) \setminus F) \geqslant \sum_{i=1}^{p}(k-|Z_i|) - |F|.$$

By dividing by 2 and rounding up the right hand, we get inequality (12). □

4. Facial aspect

In this section, we discuss the facial aspect of the kNCSP polytope. Namely, we investigate the conditions under which the inequalities presented in the previous section define facets of kNCSP(G). In the following we assume that G is $(k+1)$-node-connected. By Corollary 1, kNCSP(G) is then full-dimensional.

Grötschel and Monma (1990) characterize when the trivial inequalities define facets.

Theorem 5 Grötschel and Monma (1990).

1. *Inequality (1) defines a facet for kNCSP(G) if and only if e does not belong to a cut $\delta_{G\setminus Z}(W)$ for some $Z \subset V$ containing exactly $k+1-|Z|$ edges.*
2. *Inequalities (2) define facets for kNCSP(G) for every $e \in E$.*

The next theorem deals with conditions for the cut inequalities to define facets. Before, we give the following remark that will be helpful for proving the results below.

Remark 1. Let W and $\overline{W}$ be a partition of G such that $|W| \geqslant k, |\overline{W}| \geqslant k+1$ and $G[W]$ and $G[\overline{W}]$ are both k-node-connected. Let $\{e_1, \ldots, e_k\}$ be edges of $\delta_G(W)$ forming a matching of G. Let $S = E(W) \cup E(\overline{W}) \cup \{e_1, \ldots, e_k\}$. Then S is a solution of kNCSP(G).

Theorem 6. *The cut inequality (3), induced by a node set $W \subset V$, defines a facet for kNCSP(G) if the following hold.*

(i) *$G[W]$ and $G[\overline{W}]$ are $(k+1)$-node-connected.*
(ii) *A maximum cardinality matching M in $\delta_G(W)$ contains at least $k+1$ edges.*

Proof. Let us denote by $ax \geqslant \alpha$ the cut inequality induced by W, and let $\mathcal{F} = \{x \in kNCSP(G) | ax = \alpha\}$. Suppose there exists a facet defining inequality $bx \geqslant \beta$ such that $\mathcal{F} \subseteq F = \{x \in kNCSP(G) | bx = \beta\}$. We will prove that there is a scalar ρ such that $b = \rho a$.By ii) there exists a matching $M = \{e_1, \ldots, e_p\}$, $p \geqslant k+1$, in $\delta_G(W)$ of p edges such that $e_i = u_i v_i, i = 1, \ldots, p$, with $u_i \in W$ and $v_i \in \overline{W}$. Let $U_1 = \{u_1, \ldots, u_p\}$ and $V_1 = \{v_1, \ldots, v_p\}$. Let $T_1 = E(W) \cup E(\overline{W}) \cup \{e_1, \ldots, e_k\}$. As by i) $G[W]$ and $G[\overline{W}]$ are $(k+1)$-node-connected, by Remark 1, T_1 is a solution of kNCSP(G). We will show in what follows that the coefficients b_e are equal for all $e \in \delta_G(W)$. First we show that $b_{e_i} = b_{e_j}$ for $i,j \in \{1, \ldots, p\}$. Let $T_2 = (T_1 \setminus \{e_1\}) \cup \{e_{k+1}\}$. Clearly T_2 is a solution of kNCSP. As $x^{T_1}, x^{T_2} \in \mathcal{F}$, we have that $bx^{T_1} = bx^{T_2} = \beta$. This implies that $b_{e_1} = b_{e_{k+1}} = \rho$ for some $\rho \in \mathbb{R}$. By symmetry, it follows that $b_{e_i} = \rho$ for all $i = 1, \ldots, p$. As M has a maximum cardinality, any edge $e \in \delta(W) \setminus M$ is adjacent to M. Consider an edge $f = u_i v \in [U_1, \overline{W} \setminus V_1], i \in \{1, \ldots, p\}$. Let $T_3 = (T_1 \setminus \{e_i\}) \cup \{f\}$. As x^{T_1}, $x^{T_3} \in \mathcal{F} \subset F$, we have that $bx^{T_1} = bx^{T_3} = \beta$. This yields $b_f = b_{e_i} = \rho$. Thus $b_f = \rho$ for all $f \in [U_1, \overline{W} \setminus V_1]$. By symmetry we also have $b_f = \rho$ for all $f \in [V_1, W \setminus U_1]$. Finally consider an edge $h = u_i v_j, i,j \in \{1, \ldots, p\}$, with $i \neq j$. W.l.o.g., we suppose $i,j \leqslant k$. Consider the subset $T_4 = (T_1 \setminus \{e_i, e_j\}) \cup \{h, e_{k+1}\}$. We have that T_4 is a solution of kNCSP(G), and $x^{T_4} \in \mathcal{F} \subseteq F$. Which implies that $b_{e_i} + b_{e_j} = b_h + b_{e_{k+1}}$. As $b_{e_i} = b_{e_j} = b_{e_{k+1}} = \rho$, it follows that $b_h = \rho$. Thus we obtain that $b_e = \rho$ for all $e \in \delta_G(W)$.

We will now show that $b_e = 0$ for all $e \in E \setminus \delta_G(W)$. As $G[W]$ and $G[\overline{W}]$ are $(k+1)$-node-connected, we have that $T_5 = T_1 \setminus \{e\}$ induces a k-node-connected graph for all edge $e \in E(W) \cup E(\overline{W})$. Moreover $x^{T_5} \in \mathcal{F} \subseteq F$. Hence $b_e = 0$. Consequently, we have that $b_e = \rho$ for all $e \in \delta_G(W)$, and $b_e = 0$ for all $e \in E \setminus \delta_G(W)$. Thus $b = \rho a$. □

Corollary 2. *If the graph G is complete, the cut inequality (3) induced by $W \subset V$ is facet-defining for kNCSP(G) if $|W| \geqslant k+2$ and $|\overline{W}| \geqslant k+2$.*

The following theorems give necessary conditions and sufficient conditions for the node-cut inequalities to be facet-defining.

Theorem 7. *The node-cut inequality (4), induced by a node-cut $\delta_{G\setminus Z}(W)$ for some node sets W and Z, defines a facet for kNCSP(G) only if $|[W,Z]| \geqslant |Z|+1$ and $|[V \setminus (W \cup Z), Z]| \geqslant |Z|+1$.*

Proof. Suppose for instance that $|[W,Z]| < |Z|+1$, the case where $|[V \setminus (W \cup Z), Z]| < |Z|+1$ is similar. Thus, if $|[W,Z]| < |Z|+1$, then for any solution $x \in k$NCSP(G) we have that $-x([W,Z]) \geqslant -|Z|$, and $x(\delta_G(W)) \geqslant k$. Hence we obtain that $x(\delta_{G\setminus Z}(W)) = x(\delta_G(W)) - x([W,Z]) \geqslant k - |Z|$. In consequence, $x(\delta_{G\setminus Z}(W)) \geqslant k - |Z|$ is redundant with respect to the cut and trivial inequalities, and hence cannot define a facet. □

Theorem 8. *The node-cut inequality (4) defines a facet for kNCSP(G) if the following hold.*

(i) *$G[W]$ and $G[\overline{W}]$ are $(k+1)$-node-connected.*
(ii) *A maximum cardinality matching C in $\delta_G(W)$ contains at least $k+1$ edges such that $|C \cap [Z, W]| = |Z|$ and there exists a node in W which is not incident to the matching C and it is adjacent to all the nodes of Z.*

Proof. Let us denote by $ax \geqslant \alpha$ the node-cut inequality induced by W, and let $\mathcal{F} = \{x \in kNCSP(G) | ax = \alpha\}$. Suppose there exists a facet defining inequality $bx \geqslant \beta$ such that $\mathcal{F} \subseteq F = \{x \in kNCSP(G) | bx = \beta\}$. We will prove that there is a scalar ρ such that $b = \rho a$. By ii) there exists a matching $C = \{e_1, \ldots, e_p\}, p \geqslant k+1$, in $\delta_G(W)$, such that $e_i = u_i v_i, i = 1, \ldots, p, u_i \in W$ and $v_i \in \overline{W}$, and $e_{k-t+j} \in [W, Z], j = 1, \ldots, t$, with $|Z| = t$. Let $U_1 = \{u_1, \ldots, u_p\}$ and $V_1 = \{v_1, \ldots, v_p\}$. And let $T_1 = E(W) \cup E(\overline{W}) \cup \{e_1, \ldots, e_k\} \cup [W, Z]$. As by i), $G[W]$ and $G[\overline{W}]$ are $(k+1)$-node-connected, by Remark 1, T_1 is a solution of kNCSP(G). Hence $x^{T_1} \in \mathcal{F}$. Let $T_2 = (T_1 \setminus \{e_1\}) \cup \{e_{k+1}\}$ (Recall that $p \geqslant k+1$). Clearly, T_2 is a solution of kNCSP. As $x^{T_1}, x^{T_2} \in \mathcal{F}$, we have that $bx^{T_1} = bx^{T_2} = \beta$, implying that $b_{e_1} = b_{e_{k+1}} = \rho$ for some $\rho \in \mathbb{R}$. By symmetry, we obtain that $b_{e_i} = \rho$ for all $e_i \in C \setminus [W, Z]$. As C has a maximum cardinality, any edge $e \in \delta_{G\setminus Z}(W) \setminus C$ is adjacent to C. Consider an edge $f = u_i v \in [U_1, V \setminus (W \cup Z \cup V_1)]$. W.l.o.g., we suppose $i \in \{1, \ldots, k-t\}$. Let $T_3 = (T_1 \setminus \{e_i\}) \cup \{f\}$. Set T_3 is a solution of kNCSP(G). Moreover $x^{T_3} \in \mathcal{F} \subseteq F$. Hence $bx^{T_1} = bx^{T_3} = \beta$, implying that $b_f = b_{e_i} = \rho$. Thus $b_f = \rho$ for all $f \in [U_1, V \setminus (W \cup Z \cup V_1)]$. By symmetry, we also have that $b_f = \rho$, for all $f \in [V_1 \setminus Z, W \setminus U_1]$. Now let $h = u_i v_j, i \neq j, i,j \in \{1, \ldots, p\} \setminus \{k-t+1, \ldots, k\}$. Consider $T_4 = (T_1 \setminus \{e_i, e_j\}) \cup \{h, e_{k+1}\}$. Set T_4 is a solution of kNCSP(G). Moreover $x^{T_4} \in \mathcal{F} \subseteq F$. Hence $b_{e_i} + b_{e_j} = b_h + b_{e_{k+1}}$. As $b_{e_i} = b_{e_j} = b_{e_{k+1}} = \rho$, this implies that $b_h = \rho$. Therefore we obtain that $b_e = \rho$ for all $e \in \delta_{G\setminus Z}(W)$. Now consider an edge $e \in E(W)$, and let $T_5 = T_1 \setminus \{e\}$. As $G[W]$ is

$(k+1)$-node-connected, $G[W]\setminus\{e\}$ is k-node-connected, and hence T_5 is a solution of kNCSP. Thus $x^{T_5}\in\mathcal{F}\subseteq F$, and $bx^{T_1}=bx^{T_5}=bx^{T_1}-b_e$. Which implies that $b_e=0$. Therefore $b_e=0$ for all $e\in E(W)$. Similarly, we have that $b_e=0$ for all $e\in E(\overline{W})$. Now let $e\in[W,Z]$. Let $T_6=T_1\setminus\{e\}$. By ii), T_6 contains a matching of at least k edges in $\delta(W)$. Hence T_6 is a solution of kNCSP, which implies that $b_e=0$. Consequently, we obtain that $b_e=\rho$ for all $e\in\delta_{G\setminus Z}(W)$ and $b_e=0$ for all $e\in E\setminus\delta_{G\setminus Z}(W)$. Thus, $b=\rho a$. Which ends the proof of the theorem. □

Corollary 3. *If the graph G is complete, then the node-cut inequalities (4) are facet-defining for kNCSP(G) if* $|W|\geqslant k+2$ *and* $|\overline{W}|\geqslant k+2$.

Now, we discuss sufficient conditions for the F-node-partition and SP-node-partition inequalities to define facets of kNCSP(G).

Theorem 9. *Let* $G=(V,E)$ *be a graph and an integer* $k\geqslant 2$. *Let* $Z\subset V$. *Let* $Z_i\subset Z$, *with* $|Z_i|\leqslant k-2$, *for* $i=1,\dots,p$, *and* $\pi=(V_0,V_1,\dots,V_p)$ *be a partition of* $V\setminus Z$ *where p is odd. Suppose that the following hold.*

- **(i)** $G[Z]$ *is a complete graph.*
- **(ii)** $G[V_i]$ *is* $(k+1)$*-node-connected, for* $i=0,1,\dots,p$.
- **(iii)** *For* $i=1,\dots,p$, *there exists a subset* S_i *of* $k+1$ *edges of* $\delta(V_i)$ *such that*
  - (1) $|S_i\cap[Z_i,V_i]|=|Z_i|$ *and covering all the nodes of* Z_i,
  - (2) $|S_i\cap[V_0,V_i]|=k-|Z_i|-1$ *and covering* $k-|Z_i|-1$ *nodes of* V_0,
  - (3) $|S_i\cap[V_i,V_{i-1}]|=|S_i\cap[V_i,V_{i+1}]|=1$,

where the indices are taken modulo p. Moreover, if $|V_i|\geqslant 2$, *then* S_i *is a matching, for* $i=1,\dots,p$.

- **(iv)** *There exists a set* $U_0\subset V_0$ *such that* $|U_0|\geqslant k+1$ *and* $G[Z\cup U_0]$ *is complete.*
- **(v)** *For* $i=1,\dots,p,[V_0,V_i\cup V_{i+1}]$ *contains a set* $R_i\subseteq S_i\cup S_{i+1}$ *of* $k-|Z_i|+1$ *edges covering* $k-|Z_i|+1$ *nodes of* V_0. *Let* $F_i=S_i\cap[V_0,V_i]$, *for* $i=1,\dots,p$, *and let* $F=\bigcup_{i=1}^{p}F_i$. *Then the F-node-partition inequality (12), induced by* π *and F, defines a facet of kNCSP(G).(see Fig. 2 for an illustration for* $k=4$*)*

Proof. Remark that under these conditions we can easily see that G is $(k+1)$-node-connected, thus kNCSP(G) is full dimensional. Also by Condition iv), $|V_0|\geqslant k+1$. Let us denote the F-node-partition inequality by $ax\geqslant\alpha$ and let $\mathcal{F}=\{x\in kNCSP(G)|ax=\alpha\}$. Clearly, $\mathcal{F}$ is a proper face of kNCSP(G). Now suppose that there exists a facet defining inequality $bx\geqslant\beta$ of kNCSP(G) such that $\mathcal{F}\subseteq\mathcal{F}'=\{x\in kNCSP(G)|bx=\beta\}$. We will show that $b=\rho a$ for some scalar $\rho\in\mathbb{R}$. (See. Fig. 3)

For this, first remark that the right hand side of inequality (12) here is $\lceil\frac{p}{2}\rceil$. Let E_0 be the set of edges in $E\setminus F$ having both endnodes in the same element of π. Let $\Gamma=E_0\cup F\cup E(Z)\cup\delta(Z)$.

Let e_l be an edge of $S_l\cap[V_l,V_{l+1}], l=1,\dots,p$. For $l\in\{1,\dots,p\}$ consider the edge set

$$T_l=\Gamma\cup\left\{e_{l+2r},r=0,\dots,\frac{p-1}{2}\right\},$$

where the indices are taken modulo p. Observe that $x^{T_l}(\delta_{G\setminus Z}(\pi)\setminus F)=\frac{p+1}{2}$. Moreover, we have the following. □

Claim 1. T_l *induces a k-node-connected subgraph of G.*

Proof. Let G_l be the subgraph of G induced by T_l. First, we give the following remarks.

- (a) $|\delta_{G_l}(V_j)|=k$ for $j\in\{1,\dots,p\}\setminus\{l\}$ and $|\delta_{G_l}(V_l)|=k+1$.
- (b) $F_i\neq\emptyset$ since $|Z_i|\leqslant k-2$, for $i\in\{1,\dots,p\}$, and the graph obtained from G_l by removing subsets from $\{Z,V_1,\dots,V_p\}$ is connected,
- (c) The graph G_l^* obtained from G_l by contracting the sets V_0,V_1, ..., V_p,Z, and replacing the multiple edges by a single edge, and deleting the edges between V_i and $V_j,i\neq j,i,j=1,\dots,p$, is connected.

Let $Z'\subset V$ with $|Z'|=k-1$. We will show that the graph $G_l\setminus Z'$ is connected.

Case 1. $Z'\subset Z$ or $Z'\subset V_i$, for some $i\in\{1,\dots,p\}$. Suppose that $Z'\subset Z$. If $|Z|=|Z'|=k-1$, then by the Remark b) above, $G_l\setminus Z'$ is connected. So suppose $|Z|\geqslant k$. As $|Z'|=k-1$ and $G[Z]$ is complete, the subgraph induced by $Z\setminus Z'$ is connected. Moreover, by Condition iv), there exists at least one edge connecting $Z\setminus Z'$ to V_0. Since $G_l\setminus Z$ is connected, we obtain that $G_l\setminus Z'$ is also connected.

Now suppose $Z'\subset V_i$ for some $i\in\{1,\dots,p\}$. As $G[V_i]$ is $(k+1)$-node-connected and $|Z'|=k-1$, $G[V_i\setminus Z']$ is connected. Therefore, using Condition iii), the proof can be done along the same line.

Case 2. $Z'\subset V_0$. As $|Z'|=k-1$, by Condition iv), it follows that $[Z,V_0\setminus Z']_{G_l}\neq\emptyset$. We distinguish two cases. Suppose first that for every $s\in\{1,\dots,p\}$ such that $[V_s,V_{s+1}]_{G_l}\neq\emptyset$, at least one of the sets V_s and V_{s+1} is adjacent to Z in G_l. Then the graph obtained from G_l by removing V_0 is connected. Moreover, since $G[V_0]$ is $(k+1)$-node-connected, we have that $G[V_0\setminus Z']$ is connected. Therefore $G_l\setminus Z'$ is connected.

If this is not the case, then there is $s\in\{1,\dots,p\}$ such that $[V_s,Z]_{G_l}=\emptyset=[V_{s+1},Z]_{G_l}$ and $[V_s,V_{s+1}]_{G_l}\neq\emptyset$. We then have $Z_s=Z_{s+1}=\emptyset$. Moreover, by Condition v), it follows that there is

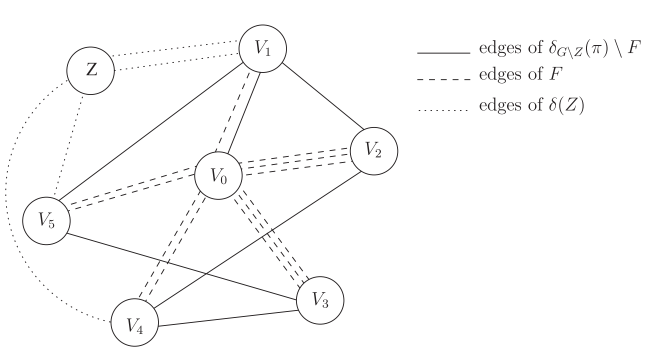

Fig. 2. A F-node-partition configuration with $k=4$.

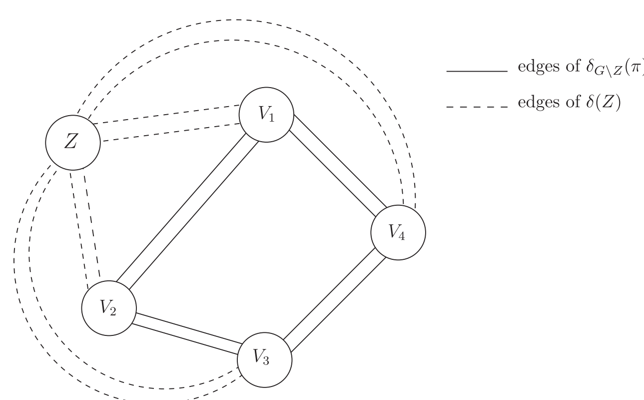

Fig. 3. An SP-node-partition configuration for $k=5$ and $|Z|=2$.

a set of $k+1$ edges between V_0 and $V_s \cup V_{s+1}$ that cover $k+1$ nodes of V_0. As $|Z'| = k-1$, at least one edge remains linking $V_0 \setminus Z'$ to $V_s \cup V_{s+1}$. Thus for $j = 1, \ldots, p$, if $[V_j, V_{j+1}]_{G_l} \neq \emptyset$, then at least one of the sets $[V_0 \setminus Z', V_j \cup V_{j+1}]_{G_l}$ and $[Z, V_j \cup V_{j+1}]_{G_l}$ is not empty. As $G[V_0]$ is $(k+1)$-node-connected and $|Z'| = k-1, G[V_0 \setminus Z']$ is connected, and hence $G_l \setminus Z'$ is connected.

Case 3. $Z' \subseteq \bigcup_{i \in I} V_i$, where $I \subset \{1, \ldots, p\}$. Note that $|I| \leqslant k-1$. Let $I' = \{i \in I || V_i| = 1\}$. First note that, by Remark b) above, the graph $G_l \setminus \bigcup_{i \in I} V_i$ is connected. Also note that as $|Z'| = k-1$, and Z' is not contained in a single set, we have $|Z' \cap V_i| \leqslant k-2$ for $i \in I \setminus I'$. Since $G[V_i]$ is $(k+1)$-node-connected, it follows that $G[V_i \setminus Z']$ is connected for $i \in I \setminus I'$. Also by construction, we have $|[V_i, V_0 \cup Z]_{G_l}| \geqslant k-1$ for $i \in I \setminus I'$. Moreover by Condition iii), $[V_i, V_0 \cup Z]_{G_l}$ contains a matching of $k-1$ edges, for $i \in I \setminus I'$. So, if no more than $k-2$ nodes are removed from V_i, at least one edge remains connecting $V_i \setminus Z'$ to $Z \cup V_0$ for $i \in I \setminus I'$. Therefore $G_l \setminus Z'$ is connected.

Case 4. $Z' \subset V_0 \cup Z \cup (\bigcup_{i \in I} V_i)$ where $I \subseteq \{1, \ldots, p\}$. Let $I' = \{i \in I || V_i| = 1\}$. Suppose first that $Z' \cap Z \neq \emptyset, Z' \cap V_0 \neq \emptyset$ and $Z' \cap \bigcup_{i \in I} V_i = \emptyset$. Consider the case where $Z \subseteq Z'$. Thus $|Z' \cap V_0| = k-1-|Z|$. By Condition v), for $i = 1, \ldots, p$, $|[V_i \cup V_{i+1}, V_0]_{G_l}| \geqslant k - |Z_i| + 1$ and covers $k - |Z_i| + 1$ nodes of V_0. As $|Z_i| \leqslant |Z|$, there exists at least one edge linking $V_0 \setminus Z'$ and $V_i \cup V_{i+1}$. As $G_l[V_0 \setminus Z']$ is connected, $G_l[(V_0 \setminus Z') \cup (\bigcup_{i=1}^{p} V_i)] = G_l \setminus Z'$ is then connected. Now suppose $Z \setminus Z' \neq \emptyset$. Since $G[V_0]$ is $(k+1)$-node-connected and $G[Z]$ is a complete graph, it follows that $G[V_0 \setminus Z']$ and $G[Z \setminus Z']$ are connected. Moreover, by Condition iv), $[Z \setminus Z', V_0 \setminus Z']_{G_l} \neq \emptyset$. Also, by construction, at least one edge remains connecting V_i to $V_{i+1} \cup V_{i-1}$, for $i = 2, \ldots, p+1$, where the indices are modulo p. By Condition iii) and v) $|[V_0 \cup Z, V_i \cup V_{i+1}]_{G_l}| \geqslant k+1$ and $[V_0 \cup Z, V_i \cup V_{i+1}]_{G_l}$ covers $k+1$ nodes of $V_0 \cup Z$, for $i = 1, \ldots, p$. As $|Z'| = k-1$, there exists at least one edge linking $(V_0 \cup Z) \setminus Z'$ to $V_i \cup V_{i+1}$, for $i = 1, \ldots, p$. Thus $G_l \setminus Z'$ is connected.

Now suppose that $Z' \cap Z \neq \emptyset, Z' \cap V_i \neq \emptyset$, for $i \in I$, and $Z' \cap V_0 = \emptyset$. If $Z \subset Z'$, the proof is similar to the previous case. Suppose that $Z \setminus Z' \neq \emptyset$. We have that $|Z' \cap Z| \leqslant k - |I| - 1$. Let $I' = \{i \in I | V_i \cap Z' \neq \emptyset$ and $|V_i| = 1\}$. Since $G[V_i]$, for $i \in I \setminus I'$, is $(k+1)$-node-connected and $G[Z]$ is complete, it follows that $G[Z \setminus Z']$, and $G[V_i \setminus Z']$, for $i \in I \setminus I'$, are connected. By Condition iv) there exists at least one edge connecting $Z \setminus Z'$ to V_0. Also by Condition iii), $[V_i, V_0 \cup Z]_{G_l}$ contains a matching of $k-1$ edges, for $i \in I \setminus I'$. So if no more than $k-2$ nodes are removed from $V_i \cup Z$, at least one edge remains connecting $V_i \setminus Z'$ to $(Z \setminus Z') \cup V_0$. Thus $G_l \setminus Z'$ is connected.

Now, if $Z' \cap V_0 \neq \emptyset$ and $Z' \cap V_i \neq \emptyset$, for $i \in I$, and $Z' \cap Z = \emptyset$, then we have that $|Z' \cap V_0| \leqslant k - |I| - 1$ and $|Z' \cap V_i| \leqslant k - |I| - 1$ for $i \in I$. Since $G[V_i]$, for $i \in \{0\} \cup (I \setminus I')$, is $(k+1)$-node-connected, it follows that $G[V_i \setminus Z']$ is connected. By Condition iv) we have that $[V_0 \setminus Z', Z] \neq \emptyset$. Moreover by Condition iii), $[V_i, V_0 \cup Z]_{G_l}$ contains a matching of $k-1$ edges, for $i = 1, \ldots, p$. So if no more than $k-2$ nodes are removed from $V_i \cup V_0$ at least one edge remains connecting $V_i \setminus Z'$ to $Z \cup (V_0 \setminus Z')$. Thus $G_l \setminus Z'$ is connected.

Suppose now that $Z' \cap Z \neq \emptyset, Z' \cap V_0 \neq \emptyset$ and $Z' \cap (\bigcup_{i \in I} V_i) \neq \emptyset$. We have that $|Z' \cap Z| \leqslant k - |I| - 2$ and $|Z' \cap V_i| \leqslant k - |I| - 2$ for $i \in I$. If $Z \subseteq Z'$, the proof is similar to a previous case. Since $G[V_i]$, for $i \in \{0\} \cup I \setminus I'$, is $(k+1)$-node-connected and $G[Z]$ is complete, it follows that $G[Z \setminus Z']$ and $G[V_i \setminus Z']$ are connected, for $i \in \{0\} \cup I \setminus I'$. By Condition iv), there exists at least one edge connecting $Z \setminus Z'$ to V_0. Also by Condition iii), there is a matching of $k-1$ edges between V_i and $Z \cup V_0$. We claim that $[V_i \cup V_{i+1}, Z \cup V_0]_{G_l \setminus Z'} \neq \emptyset$, for $i = 1, \ldots, p$, where the indices are taken modulo p. This is clear if $[V_i, Z \cup V_0]_{G_l \setminus Z'} \neq \emptyset$. So suppose $[V_i, Z \cup V_0]_{G_l \setminus Z'} = \emptyset$, for some $i_0 \in I \setminus I'$. As $[V_{i_0}, Z \cup V_0]_{G_l}$ contains a matching of $k-1$ edges and $|Z'| = k-1$, this implies that $Z' \subset (Z \cup V_0 \cup V_{i_0})$. Thus $|I| = 1$ and $|I'| = 0$. Moreover, $Z' \cap V_i = \emptyset$ for $i \in \{1, \ldots, p\} \setminus \{i_0\}$. Since $[V_i, Z \cup V_0]_{G_l}$ contains a matching of $k-1$ edges for $i \in \{1, \ldots, p\} \setminus \{i_0\}$, and no more than $k-2$ nodes are removed from $V_0 \cup Z$, at least one edge remains connecting $(V_0 \cup Z) \setminus Z'$ to V_i. Thus $[V_i \cup V_{i+1}, Z \cup V_0]_{G_l \setminus Z'} \neq \emptyset$, and the claim is proved. As a consequence, we obtain that $G_l \setminus Z'$ is connected. Hence $G_l = (V, T_l)$ is k-node-connected.

Thus $x^{T_l} \in \mathcal{F}$. □

Now we show that $b(e) = \rho a(e)$ for all $e \in E \setminus \Gamma$, for some $\rho \in \mathbb{R}$.

As $x^{T_1}, \ldots, x^{T_p}$ belong to $\mathcal{F}$, it follows that $bx^{T_1} = \ldots = bx^{T_p} = \beta$. Hence $b(e_1) = \ldots = b(e_p)$. As e_l is an arbitrary edge of $S_l \cap [V_l, V_{l+1}]$, $l = 1, \ldots, p$, respectively, we obtain

$b(e) = \rho$ for all $e \in S_l \cap [V_l, V_{l+1}]$, $l = 1, \ldots, p$, for some $\rho \in \mathbb{R}$.

Let r_l be an edge of $[V_l, V_{l+1}] \setminus S_l$, for $l = 1, \ldots, p$. Consider the edge set $T'_l = T_l \setminus \{e_{l+1}\} \cup \{r_l\}$. As r_l is an arbitrary edge of $[V_l, V_{l+1}] \setminus S_l$, $l = 1, \ldots, p$, respectively, we obtain

$b(e) = \rho$ for all $e \in [V_l, V_{l+1}] \setminus S_l$, $l = 1, \ldots, p$, for some $\rho \in \mathbb{R}$.

Let g_{l+1} be a fixed edge of $[V_{l+1}, V_0] \setminus F$, for $l \in \{0, \ldots, p-1\}$. Consider the edge set

$\widetilde{T}_l = (T_l \setminus \{e_l\}) \cup \{g_{l+1}\}$.

Similary, we can show that $\widetilde{T}_l$ induces a k-node-connected subgraph of G. As x^{T_l} and $x^{\widetilde{T}_l}$ belong to $\mathcal{F}$, it follows in a similar way that $b(e_l) = b(g_{l+1})$. As $b(e_l) = b(e_{l+1}) = \rho$, this yields $b(g_{l+1}) = \rho$. By exchanging the roles of V_{l+1} and $V_l, l = 1, \ldots, p$, we obtain that $b(e) = \rho$ for all $e \in \delta_G(V_0) \setminus F$. In consequence, the $b(e)$, for all $e \in E \setminus \Gamma$ have the same value ρ.

Next, we will show that $b(e) = 0$ for all $e \in \Gamma$.

Note that there are $k+1$ edges incident to V_l in the graph induced by T_l. By using Condition iii) we can show in a similar way as in the claim above that for any edge $e \in F_l, l \in \{0, \ldots, p\}, T^*_l = T_l \setminus \{e\}$ also induces a k-node-connected subgraph of G. As x^{T_l} and $x^{T^*_l}$ belong to $\mathcal{F}$, it follows that $bx^{T_l} = bx^{T^*_l} = \beta$, implying that $b(e) = 0$ for all $e \in F_l$. As l is arbitrarily chosen, we obtain that $b(e) = 0$ for all $f \in F$. Moreover, as the subgraphs induced by $V_0, \ldots, V_p$ are all $(k+1)$-node-connected, the subgraph induced by $T_l \setminus \{e\}$, for all $e \in E_0$, is k-node-connected. This yields as before $b(e) = 0$ for all $e \in E_0$.

Now suppose that $e \in E(Z)$. By Conditions i) and iv) we can clearly see that $T_l \setminus \{e\}$ also induces a k-node-connected subgraph of G. Implying that $b(e) = 0$.

Let h be an edge of $\delta(Z)$. We can show in a similar way as in the claim above that $\overline{T}_l = T_l \setminus \{h\}$ also induces a k-node-connected subgraph of G. As $x^{\overline{T}_l}$ belongs to $\mathcal{F}$, it follows that $b(h) = 0$. Consequently $b(e) = 0$ for all $e \in \Gamma$.

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

Corollary 4. *If the graph G is complete, then the F-node-partition inequalities (4) are facet-defining for kNCSP(G) if either* $|V_i| = 1$ *or* $|V_i| \geqslant k+2$, *for* $i = 1, \ldots, p$, *and* $|V_0| \geqslant k+2$.

Corollary 5. *Suppose that* $V_i = \{u_i\}$, *that is* $|V_i| = 1$, *for all* $i \in \{1, \ldots, p\}, |[V_0, V_i]| = k - |Z_i| - 1, |[V_i, Z_i]| = |Z_i|$, *and the nodes* $u_1, \ldots, p_p$ *form an odd cycle C. Also, suppose that* $G[V_0]$ *is* $(k+1)$-*node-connected and* $G[V_0 \cup Z]$ *is complete. Then the inequality*

$$x(C) \geqslant \left\lceil \frac{p}{2} \right\rceil$$

is valid for kNCSP(G), and defines a facet.

A graph is said to be *outerplanar* if it can be drawn in the plane without crossings in such a way that all of the vertices belong to the unbounded face of the drawing. That is, no vertex is totally surrounded by edges.

Theorem 10 Diarrassouba and Slama (2007). *Let $G = (V, E)$ be a graph and an integer $k \geqslant 2$. Let $Z \subset V$, such that $|Z| \leqslant k-1$ and $k-|Z|$ is odd. Let $\pi = (V_1, \ldots, V_p)$ be a partition of $V \setminus Z$ such that $(G \setminus Z)_\pi$ is series-parallel. Then the SP-node-partition inequality (10) associated with π defines a facet of kNCSP(G) only if $(G \setminus Z)_\pi$ is outerplanar and 2-node-connected.*

Theorem 11. *Let $G = (V, E)$ be a graph and an integer $k \geqslant 2$. Let $Z \subset V$, such that $|Z| \leqslant k-1$ and $k-|Z|$ is odd. Let $\pi = (V_1, \ldots, V_p)$ be a partition of $V \setminus Z$ such that $(G \setminus Z)_\pi$ is series-parallel. Then the SP-node-partition inequality (10) associated with π defines a facet of kNCSP(G) if the following conditions hold.*

- **i**) *$(G \setminus Z)_\pi$ is outerplanar and 2-node-connected.*
- **ii**) *$p \geqslant k+1$.*
- **iii**) *$G[V_i]$ is $(k+1)$-node-connected, for $i = 1, \ldots, p$.*
- **iv**) *$G[Z]$ is complete.*
- **v**) *For $i = 1, \ldots, p$, there exists a subset $S_i \subset \delta(V_i)$, with $|S_i| = k+1$, S_i is a matching and such that*
  - a) $|S_i \cap [Z, V_i]| = |Z|$,
  - b) $|S_i \cap [V_i, V_{i-1}]| = |S_i \cap [V_i, V_{i+1}]| = \left\lceil \frac{k-|Z|}{2} \right\rceil$,

where the indices are taken modulo p.

Proof. Let us denote the SP-node-partition inequality by $ax \geqslant \alpha$ and let $\mathcal{F} = \{x \in kNCSP(G) | ax = \alpha\}$. We start the proof by showing that, under Conditions i)–v), the polytope kNCSP(G) is full dimensional, and that $\mathcal{F} \neq \emptyset$, that is $\mathcal{F}$ is a proper face of kNCSP(G).

Let $E_0 = \bigcup_{i=1}^p E(V_i)$ and let $F_i = S_i \cap [V_i, V_{i+1}]$, for $i = 1, \ldots, p$. Note that by v) b) and the fact that $k-|Z|$ is odd, we have that $|F_i| = \frac{k-|Z|+1}{2}$, for all $i \in \{1, \ldots, p\}$. The indices are taken modulo p. Now consider the edge set $T_0 = (\bigcup_{i=1}^p F_i) \cup E_0 \cup E(Z) \cup \delta_G(Z)$. We will show that the graph induced by T_0 is $(k+1)$-node-connected. □

Claim 2. *The subgraph H induced by T_0 is $(k+1)$-node-connected.*

Proof. Since $(G \setminus Z)_\pi$ is outerplanar and 2-node-connected, its nodes are on an elementary cycle. Let H denote the graph induced by T_0. We first show that H is $(k+1)$-edge-connected. For this, let π_0 be a partition $(V_1, \ldots, V_p, Z)$ of V. We can notice that the graph H_{π_0}, induced by π_0, is a wheel, with multiple edges, whose central node z, that is the node of H_{π_0} corresponding to Z, and border nodes $v_1, \ldots, v_p$, that are the nodes corresponding to node sets $V_i, i = 1, \ldots, p$. Clearly, $|\delta_{H_{\pi_0}}(v_i)| = k+1$, for all $i = 1, \ldots, p$, and $|\delta_{H_{\pi_0}}(z)| = p|Z|$. From this, we can observe that H_{π_0} is $(k+1)$-edge-connected. As $p \geqslant k+1$ and $G[Z]$ is complete, by v) a), it follows that $\delta_H(U) \geqslant k+1$ for all $U \subset Z$. Moreover, since in addition subgraphs $G[V_i]$, for all $i \in \{1, \ldots, p\}$, are $(k+1)$-node-connected, and hence $(k+1)$-edge-connected, we get that H is $(k+1)$-edge-connected.

Now to show that H is $(k+1)$-node-connected, it suffices to show that for all $Z' \subseteq V$ with $|Z'| = k$, the graph $H \setminus Z'$ is connected, that is $|\delta_{H \setminus Z'}(W)| \geqslant 1$, for all $W \subseteq V \setminus Z'$. Thus, let $Z' \subseteq V$ with $|Z'| = k$ and $W \subseteq V \setminus Z'$. Suppose first that W crosses at least one node set $V_i \setminus Z'$, with $i \in \{1, \ldots, p\}$, that is $W \cap (V_i \setminus Z') \neq \emptyset \neq (V_i \setminus Z') \setminus W$. Since, $G[V_i]$ is $(k+1)$-node-connected, we have that $G[V_i] \setminus Z'$ is connected, and hence, $|[W \cap (V_i \setminus Z'), (V_i \setminus Z') \setminus W]| \geqslant 1$. Moreover, $[W \cap (V_i \setminus Z'), (V_i \setminus Z') \setminus W] \subseteq \delta_{H \setminus Z'}(W)$, which implies that $|\delta_{H \setminus Z'}(W)| \geqslant 1$.

Suppose now that W does not cross any node set $V_i \setminus Z', i = 1, \ldots, p$. In this case, we have that $\delta_{H \setminus Z'}(W) \subseteq E_{\pi_0} \setminus \delta_{H_{\pi_0}}(Z')$, where E_{π_0} is the set of edges of H_{π_0}. We can easily see that $E_{\pi_0} = \bigcup_{i=1}^p S_i$, and that, since the edge sets $S_i, i = 1, \ldots, p$, are matchings, removing the k nodes of Z' from G yields removing at most k edges from H_{π_0}. Thus, if L denotes the set of those edges, we have that $|\delta_{H \setminus Z'}(W)| \geqslant |\delta_{H_{\pi_0}}(W)| - |L| \geqslant |\delta_{H_{\pi_0}}(W)| - k$. Since, as shown before, H_{π_0} is $(k+1)$-edge-connected, we have that $|\delta_{H_{\pi_0}}(W)| \geqslant k+1$, and obtain $|\delta_{H \setminus Z'}(W)| \geqslant 1$.

Therefore, the graph $H \setminus Z'$ is connected, yielding that H is $(k+1)$-node-connected. □

As T_0 induces a spanning subgraph of G and is $(k+1)$-node-connected, G is also $(k+1)$-node-connected, and hence kNCSP(G) is full dimensional. Moreover, as a consequence, it follows that $T_i = T_0 \setminus \{f_i\}$, for some $f_i \in F_i, i \in \{1, \ldots, p\}$, induces a solution of the kNCSP(G). Moreover, its incidence vector satisfies with equality the SP-node-partition inequality induced by π and Z. Thus, $\mathcal{F}$ is a proper face of kNCSP(G). Now, suppose that there exists a facet-defining inequality $bx \geqslant \beta$ of kNCSP(G) such that $\mathcal{F} \subseteq \mathcal{F}' = \{x \in kNCSP(G) | bx = \beta\}$. We will show that $b = \rho a$ for some scalar $\rho \in \mathbb{R}$.

First, let $i \in \{1, \ldots, p\}$, and consider the edge set $T_{i+1} = (T_i \setminus \{f_{i+1}\}) \cup \{f_i\}$, where f_i and f_{i+1} are edges of F_i and F_{i+1}, respectively. The indices are taken modulo p. As before, we can see that T_{i+1} induces a solution of the kNCSP and its incidence vector satisfies $ax^{T_{i+1}} = \alpha = ax^{T_i}$. Thus, $bx^{T_{i+1}} = bx^{T_i} = \beta$, and hence $b(f_i) = b(f_{i+1})$. As f_i and f_{i+1} are arbitrary edges of F_i and F_{i+1}, respectively, it follows that $b(e) = b(e')$ for all $e \in F_i$ and $e' \in F_{i+1}$, for all $i \in \{1, \ldots, p\}$.

Now let r_i be an edge of $[V_i, V_{i+1}] \setminus F_i$, for some $i \in \{1, \ldots, p\}$, and consider the edge set $T'_i = T_i \setminus \{f_{i+1}\} \cup \{r_i\}$, with $f_{i+1} \in F_{i+1}$. Using similar arguments as in Claim 2, we can show that T'_i induces a solution of the kNCSP. Moreover, its incidence vector satisfies $ax^{T'_i} = \alpha$. Thus, $x^{T'_i} \in \mathcal{F}'$, and hence, $bx^{T'_i} = bx^{T_i} = \beta$. This implies that $b(r_i) = b(f_{i+1})$. Since, r_i is an arbitrary edge of $[V_i, V_{i+1}] \setminus F_i$, we have that

$$b(e) = \rho, \text{ for all } e \in [V_i, V_{i+1}], i \in \{1, \ldots, p\}, \text{ and some } \rho \in \mathbb{R}.$$

Now let $h \in [V_i, V_j]$, for some $i, j \in \{1, \ldots, p\}$ with $|i-j| > 1$, and consider the edge set $T''_i = (T_i \setminus \{f_{i-1}\}) \cup \{h\}$. As before, one can see that T''_i induces a solution of kNCSP, and its incidence vector satisfies $ax^{T''_i} = \alpha$. Thus, $x^{T''_i} \in \mathcal{F}'$ and $bx^{T''_i} = bx^{T_i} = \beta$, which implies that $b(h) = b(f_{i-1})$. Therefore,

$$b(e) = \rho, \text{ for all } e \in E_\pi. \tag{13}$$

Finally, we show that $b(e) = 0$, for all $e \in E_0 \cup E(Z) \cup \delta_G(Z)$. For this, consider an edge $e \in E(V_i)$, for some $i \in \{1, \ldots, p\}$. Since $G[V_i]$ is $(k+1)$-node-connected, the edge set $T_i \setminus \{e\}$ clearly induces a solution of the kNCSP and its incidence vector satisfies $ax \geqslant \alpha$ with equality. Thus, $bx^{T_i \setminus \{e\}} = bx^{T_i} = \beta$, implying $b(e) = 0$. Now, for an edge $e \in E(Z)$, by Conditions ii) and iv), we can clearly see that $T_i \setminus \{e\}$, for some $i \in \{1, \ldots, p\}$, induces a solution of the kNCSP. Also, its incidence vector satisfies with equality $ax \geqslant \alpha$, and, as before, $b(e) = 0$. Finally, let g be an edge of $\delta_G(Z)$. As before, the edge set $T_i \setminus \{g\}$, for some $i \in \{1, \dots, p\}$, induces a solution of the kNCSP, and its incidence vector satisfies with equality $ax \geqslant \alpha$. It follows that

$$b(e) = \rho, \text{ for all } e \in e \in E_0 \cup E(Z) \cup \delta_G(Z). \quad (14)$$

Consequently, from (13) and (14), we obtain that $b = \rho a$, which ends the proof.

5. Structural properties

In this section, we discuss some structural properties of the extreme points of the linear relaxation $P(G,k)$ of the kNCSP polytope. Recall that $P(G,k)$ is the polytope given by inequalities (1)–(4).

For this, we first give some notations and definitions. Let $\bar{x} \in P(G,k)$ be a solution. We say that an inequality $ax \geqslant \alpha$ is *tight* for $\bar{x}$ if $a\bar{x} = \alpha$. We will denote by $E_0(\bar{x}), E_1(\bar{x})$ and $E_f(\bar{x})$, the following edge sets

- $E_0(\bar{x}) = \{e \in E \mid \bar{x}(e) = 0\}$,
- $E_1(\bar{x}) = \{e \in E \mid \bar{x}(e) = 1\}$,
- $E_f(\bar{x}) = \{e \in E \mid 0 < \bar{x}(e) < 1\}$.

Also we let $\mathcal{C}_{PE}(\bar{x})$ (resp. $\mathcal{C}_{PN}(\bar{x})$) be the set of cuts $\delta(W)$ (resp. node-cuts $\delta_{G\setminus Z}(W)$) that are tight for $\bar{x}$. If $\bar{x}$ is an extreme point of $P(G,k)$, then $\bar{x}$ is the unique solution of the linear system

$$S(\bar{x})\begin{cases} x(e) = 0, & \text{for all } e \in E_0(\bar{x}),\\ x(e) = 1, & \text{for all } e \in E_1(\bar{x}),\\ x(\delta_G(W)) = k, & \text{for all cuts } \delta_G(W) \in \mathcal{C}^*_{PE}(\bar{x}),\\ x(\delta_{G\setminus Z}(W)) = k - |Z|, & \text{for all node-cuts } \delta_{G\setminus Z}(W) \in \mathcal{C}^*_{PN}(\bar{x}),\end{cases}$$

where $\mathcal{C}^*_{PE}(\bar{x})$ (resp. $\mathcal{C}^*_{n}(\bar{x})$) is a subset of $\mathcal{C}_{PE}(\bar{x})$ (resp. $\mathcal{C}_{PN}(\bar{x})$).

Lemma 1. *Let $\bar{x} \in P(G,k)$ and $W \subseteq V$ such that the cut induced by W is tight for $\bar{x}$.*

1. *If for some $R \subset V, \bar{x}(\delta(R)) = k$, then $\bar{x}(\delta(W \cap R)) = k$ and $\bar{x}(\delta(W \cup R)) = k$.*
2. *If for some $Z \subset V$ such that $|Z| \leqslant k-1$ and $Z \cap W \neq \emptyset \neq Z \cap \overline{W}$, and for $T \subset V \setminus Z$, such that $T \cap W \neq \emptyset, T \not\subset W, W \not\subset T$ and $T \cup W \neq V \setminus Z$, we have $\bar{x}(\delta_{G\setminus Z}(T)) = k - |Z|$, then $\bar{x}(\delta_{G\setminus (Z\cap W)}(W \cap T)) = k - |Z \cap W|$, $\bar{x}(\delta_{G\setminus (Z\cap \overline{W})}(W \cup T)) = k - |Z \cap \overline{W}|$, $\bar{x}(\delta_{G\setminus (Z\cap \overline{W})}(\overline{W} \cap T)) = k - |Z \cap \overline{W}|$, $\bar{x}(\delta_{G\setminus (Z\cap W)}(\overline{W} \cup T)) = k - |Z \cap W|$.*

Proof.

1. The proof is similar to that of Cornuéjols, Fonlupt, and Naddef (1985).
2. Let $T_1 = T \cap W, T_2 = T \cap \overline{W}, Z_1 = Z \cap W, Z_2 = Z \cap \overline{W}, T_3 = W \setminus (T \cup Z_1)$ and $T_4 = \overline{W} \setminus (T \cup Z_2)$. Thus $T_i \neq \emptyset$ for $i = 1, \dots, 4$. As $\delta(W) \in \mathcal{C}_P E(\bar{x})$, we have that

$$\begin{aligned} k = \bar{x}(\delta(W)) &= \bar{x}(\delta(T_1,T_2)) + \bar{x}(\delta(T_1,T_4)) + \bar{x}(\delta(T_3,T_2)) \\ &\quad + \bar{x}(\delta(T_3,T_4)) + \bar{x}(\delta(T_1,Z_2)) + \bar{x}(\delta(T_3,Z_2)) \\ &\quad + \bar{x}(\delta(T_2,Z_1)) + \bar{x}(\delta(T_4,Z_1)) + \bar{x}(\delta(Z_1,Z_2)). \end{aligned} \quad (15)$$

And as $\delta_{G\setminus Z}(T) \in \mathcal{C}_n(\bar{x})$, we have that

$$\begin{aligned} k - |Z| = \bar{x}(\delta_{G\setminus Z}(T)) &= \bar{x}(\delta(T_1,T_3)) + \bar{x}(\delta(T_1,T_4)) + \bar{x}(\delta(T_2,T_3)) \\ &\quad + \bar{x}(\delta(T_2,T_4)). \end{aligned} \quad (16)$$

By considering the node-cuts $\delta_{G\setminus Z_1}(T_1), \delta_{G\setminus Z_2}(T_2), \delta_{G\setminus Z_1}(T_3)$ and $\delta_{G\setminus Z_2}(T_4)$, we have that

$$\begin{aligned} k - |Z_1| \leqslant \bar{x}(\delta_{G\setminus Z_1}(T_1)) &= \bar{x}(\delta(T_1,T_2)) + \bar{x}(\delta(T_1,T_3)) + \bar{x}(\delta(T_1,T_4)) \\ &\quad + \bar{x}(\delta(T_1,Z_2)), \end{aligned} \quad (17)$$

$$\begin{aligned} k - |Z_2| \leqslant \bar{x}(\delta_{G\setminus Z_2}(T_2)) &= \bar{x}(\delta(T_2,T_1)) + \bar{x}(\delta(T_2,T_3)) + \bar{x}(\delta(T_2,T_4)) \\ &\quad + \bar{x}(\delta(T_2,Z_1)), \end{aligned} \quad (18)$$

$$\begin{aligned} k - |Z_1| \leqslant \bar{x}(\delta_{G\setminus Z_1}(T_3)) &= \bar{x}(\delta(T_3,T_1)) + \bar{x}(\delta(T_3,T_2)) + \bar{x}(\delta(T_3,T_4)) \\ &\quad + \bar{x}(\delta(T_3,Z_2)), \end{aligned} \quad (19)$$

$$\begin{aligned} k - |Z_2| \leqslant \bar{x}(\delta_{G\setminus Z_2}(T_4)) &= \bar{x}(\delta(T_4,T_1)) + \bar{x}(\delta(T_4,T_2)) + \bar{x}(\delta(T_4,T_3)) \\ &\quad + \bar{x}(\delta(T_4,Z_1)). \end{aligned} \quad (20)$$

By adding (15) and (16) on one side and (18) and (19) (resp. (17) and (20)) on the other side, and comparing the two resulting constraints, as $\bar{x}(e) \geqslant 0$ for all $e \in E$, we obtain that (17) and (20) (resp. (18) and (19)) must be tight and $\bar{x}(\delta(T_1,Z_2)) = \bar{x}(\delta(T_1,T_4)) = \bar{x}(\delta(T_4,Z_1)) = \bar{x}(\delta(Z_1,Z_2)) = 0$ (resp. $\bar{x}(\delta(T_3,T_2)) = \bar{x}(\delta(T_3,Z_2)) = \bar{x}(\delta(T_2,Z_1)) = 0$), which ends the proof. □

From Lemma 1, we can show the following result. Its proof is omitted since it follows the same lines as a similar result in Cornuéjols et al. (1985).

Lemma 2. *Let $\bar{x}$ be an extreme point of $P(G,k)$, and $W \subset V$ such that $\bar{x}(\delta(W)) = k$. Then the system $S(\bar{x})$ can be chosen so that*

1. *a cut $\delta(R) \in \mathcal{C}^*_{PE}(\bar{x})$ is such that $R \subset W$ or $R \subset \overline{W}$;*
2. *a node-cut $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}(\bar{x})$ is such that $(T \cup Z) \subset W, (T \cup Z) \subset \overline{W}$, $T \subset W$ and $Z \subset \overline{W}$, or $T \subset \overline{W}$ and $Z \subset W$.*

6. Reduction operations

In this section we introduce some reduction operations defined with respect to a solution $\bar{x}$ of $P(G,k)$. These operations will be considered in a preprocessing phase for separating violated inequalities in our Branch-and-Cut algorithm. Let $\theta_1, \theta_2, \theta_3$ and θ_4 be the reduction operations defined as follows.

- 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, $\bar{x}(e) = 1$ for all $e \in E(W)$ and $\bar{x}(\delta(W)) = k$.
- θ_3: Contract a node subset $W \subseteq V$ such that $|W| \geqslant k, |\overline{W}| \geqslant k, \bar{x}(e) = 1$ for all $e \in E(W)$, and $|\delta_G(W)| = k$.
- θ_4: Replace a set of parallel edges by only one edge.

We have the following results.

Lemma 3. *Let $G' = (V, E')$ be the graph obtained from G by the application of Operation θ_1 and let $\bar{x}'$ be the restriction of $\bar{x}$ to G'. Then $\bar{x}'$ is an extreme point of $P(G',k)$ if and only if $\bar{x}$ is an extreme point of $P(G,k)$.*

Proof. Easy. □

Lemma 4. *Let $G' = (V', E')$ and $\bar{x}'$ be the graph and the solution obtained from G and $\bar{x}$ by the application of Operation θ_2. Suppose that*

(1) *$\bar{x}' \in P(G',k)$,*
(2) *for all $Z \subset W, |Z| \leqslant k-1, \delta_{G\setminus Z}(T) \notin \mathcal{C}_{PN}(\bar{x})$ for all $T \subseteq \overline{W}$.*

Then $\overline{x}'$ is an extreme point of $P(G',k)$ if $\overline{x}$ is an extreme point of $P(G,k)$.

Proof. Let W be a node set of G contracted by Operation θ_2. As $\delta(W) \in \mathcal{C}_{PE}(\overline{x})$, by Lemma 2, the system $S(\overline{x})$ can be chosen in such a way that for every $\delta(R) \in \mathcal{C}^*_{PE}(\overline{x})$ (resp. $\delta_{G\setminus Z_T}(T) \in \mathcal{C}^*_{PN}(\overline{x})$) either $R \subseteq W$ or $R \subseteq \overline{W}$ (resp. $(T \cup Z_T) \subset W, T \subset W$ and $Z_T \subset \overline{W}$, or $T \subset \overline{W}$ and $Z_T \subset W$). As $\overline{x}(e) = 1$ for all $e \in E(W)$ and $G[W]$ is k-edge connected, this implies that $\mathcal{C}^*_{PE}(\overline{x}) \subseteq \mathcal{C}_{PE}(\overline{x}')$. Moreover by (2) it follows that if $\delta_{G\setminus Z_T}(T)$ is tight for $\overline{x}$ and $Z_T \subseteq W$, then $W \cap T \neq \emptyset$ and $W \setminus (Z_T \cup T) \neq \emptyset$. Let $T_1 = W \cap T$ and $T_2 = W \setminus (Z_T \cup T)$. We have that $k - |Z_T| = \overline{x}(\delta_{G\setminus Z_T}(T)) \geqslant \overline{x}(\delta(T_1,T_2)) \geqslant k$, a contradiction. The last inequality comes from the fact that $G[W]$ is k-edge connected and $\overline{x}(e) = 1$ for all $e \in E(W)$. Inconsequence, all the node-cuts $\delta_{G\setminus Z_T}(T)$ of $\mathcal{C}^*_{PN}(\overline{x})$ are such that $Z_T \subset \overline{W}$. However these are at the same time tight for $\overline{x}'$. Thus $\mathcal{C}^*_{PN}(\overline{x}) \subset \mathcal{C}_{PN}(\overline{x}')$. Let $S'(\overline{x})$ be the system obtained from $S(\overline{x})$ by deleting the equations $x(e) = 1$ for all $e \in E(W)$. Then $\overline{x}'$ is the unique solution of $S'(\overline{x})$. As all the equations of $S'(\overline{x})$ come from $P(G',k)$ and by 1) $\overline{x}' \in P(G',k)$, it follows that $\overline{x}'$is an extreme point of $P(G',k)$. □

Lemma 5. *Let $G' = (V',E')$ and $\overline{x}'$ be the graph and the solution obtained from G and $\overline{x}$, respectively, by the application of Operation θ_3. Then $\overline{x}'$ is an extreme point of $P(G',k)$.*

Proof. Let $W \subseteq V$ be a node set satisfying the conditions of Operation θ_3. First observe that as $|\delta(W)| = k$, we have that $\overline{x}(e) = 1$ for all $e \in \delta(W)$ and $\overline{x}(\delta(W)) = k$. Thus, by Lemma 2, $S(\overline{x})$ can be chosen so that for every node-cut $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}$, we have $(T \cup Z) \subset W$, $(T \cup Z) \subset \overline{W}, T \subset W$ and $Z \in \overline{W}$, or $T \subset \overline{W}$ and $Z \in W$. We will show that any cut $\delta(R) \in \mathcal{C}^*_{PE}(\overline{x})$ is such that $R \subset \overline{W}$, and any node-cut $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}(\overline{x})$ is such that $(T \cup Z) \subset \overline{W}$. Suppose the contrary and consider first that for some $\delta(R) \in \mathcal{C}^*_{PE}(\overline{x}), R \subsetneq W$. As $\overline{x}(e) = 1$, for all $e \in E(W) \cup \delta(W)$, one can see that $|\delta(R)| = k$, and hence $\overline{x}(\delta(R)) = k$ can be obtained from $\overline{x}(e) = 1$, for all $e \in \delta(R)$, contradicting the fact that $\delta(R) \in \mathcal{C}^*_{PE}(\overline{x})$. Now suppose that for some node-cut $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}(\overline{x})$ either $(T \cup Z) \subset W$ or $T \subseteq W$ and $Z \subseteq \overline{W}$. We can show similarly to the previous case that $|\delta_{G\setminus Z}(T)| = k - |Z|$ and that $\overline{x}(\delta_{G\setminus Z}(T)) = k - |Z|$ can be obtained from $\overline{x}(e) = 1$, for all $e \in \delta_{G\setminus Z}(T)$, which yields a contradiction.

We consider now a node-cut $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}(\overline{x})$ such that $T \subseteq \overline{W}$ and $Z \subseteq W$. Notice that, as $|W| \geqslant k$, we have that $W \setminus Z \neq \emptyset$. If $T = \overline{W}$, then $\overline{x}(\delta_{G\setminus Z}(T)) = \overline{x}(\delta(W \setminus Z, T)) = |\delta(W \setminus Z, T)| = k - |Z|$. Thus, $\overline{x}(\delta_{G\setminus Z}(T)) = k - |Z|$ can be obtained from the equations $\overline{x}(e) = 1$, for all $e \in \delta_{G\setminus Z}(T)$, contradicting the fact that $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}(\overline{x})$. Thus, $\overline{W} \setminus T \neq \emptyset$. For convenience, we let $T_1 = W \setminus Z$ and $T_2 = \overline{W} \setminus T$. First, note that

$$\overline{x}(\delta_{G\setminus Z}(T)) = \overline{x}(\delta(T,T_1)) + \overline{x}(\delta(T,T_2)) = k - |Z|. \tag{21}$$

Eq. (21) together with the cut inequality induced by T yields

$$\overline{x}(\delta(T,Z)) \geqslant |Z|. \tag{22}$$

Also, as by the assumption $|\delta(W)| = k$, we have that

$$\overline{x}(\delta(T,T_1)) + \overline{x}(\delta(T,Z)) + \overline{x}(\delta(T_2,T_1)) + \overline{x}(\delta(T_2,Z)) = k. \tag{23}$$

This equation, together with the node-cut inequality induced by $\delta_{G\setminus Z}(T_2)$ implies that

$$\overline{x}(\delta(T,Z)) + \overline{x}(\delta(T_2,Z)) \leqslant |Z|. \tag{24}$$

Thus, by inequalities (22) and (24), we have that $\overline{x}(\delta(T,Z)) = |Z|$ and $\overline{x}(\delta(T_2,Z)) = 0$, and hence

$$\overline{x}(\delta(T)) = \overline{x}(\delta(T,T_1)) + \overline{x}(\delta(T,Z)) + \overline{x}(\delta(T,T_2)) = k. \tag{25}$$

Moreover, as $\overline{x}(e) = 1$, for all $e \in \delta(W)$, we have that $\overline{x}(\delta(T,Z)) = |Z| = |\delta(T,Z)|$. Therefore, $\overline{x}(\delta_{G\setminus Z}(T)) = k - |Z|$ can be obtained from (25) and the $\overline{x}(e) = 1$, for all $e \in \delta(T,Z)$, and hence, can be replaced in $S(\overline{x})$ by Eq. (25).

Consequently, the system $S(\overline{x})$ can be chosen so that $R \subseteq \overline{W}$ for every cut $\delta(R) \in \mathcal{C}^*_{PE}(\overline{x})$ and $T \cup Z \subseteq \overline{W}$ for every node-cut $\delta_{G\setminus Z}(T) \in \mathcal{C}^*_{PN}(\overline{x})$. This also implies that $\mathcal{C}^*_{PE}(\overline{x}) \cup \mathcal{C}^*_{PN}(\overline{x}) \subseteq \mathcal{C}^*_{PE}(\overline{x}') \cup \mathcal{C}^*_{PN}(\overline{x}')$. Thus, $\overline{x}'$ is the unique solution of a subsystem of $S(\overline{x})$. As all the equations of that subsystem correspond to constraints of $P(G \setminus W, k)$, this implies that $\overline{x}'$ is an extreme point of $P(G \setminus W, k)$. □

Lemma 6. *Let $G' = (V',E')$ be the graph obtained from G by the application of Operation θ_4. Let E_0 be the set of parallel edges of G and e_0 the edge replacing E_0 in G'. Let $\overline{x}'$ be the solution given by $\overline{x}'(e) = \overline{x}(e)$ if $e \in E \setminus E_0$ and $\overline{x}'(e) = 1$ if $e = e_0$. Then $\overline{x}'$ is an extreme point of $P(G',k)$.*

Proof. Observe that for every cut $\delta(W)$ (node-cut $\delta_{G\setminus Z}(W)$) either $E_0 \subseteq \delta(W)$ $(E_0 \subset \delta_{G\setminus Z}(W))$ or $E_0 \cap \delta(W) = \emptyset$ $(E_0 \cap \delta_{G\setminus Z}(W) = \emptyset)$. Moreover, E_0 cannot contain more than two edges with fractional value. Indeed, if $e_1, e_2 \in E_0$ and $0 < x(e_1) < 1$ and $0 < x(e_2) < 1$, let $\overline{x}^*$ be the solution given by $\overline{x}^*(e) = \overline{x}(e)$ if $e \in E \setminus \{e_1,e_2\}, \overline{x}^*(e) = \overline{x}(e) + \epsilon$ if $e = e_1$ and $\overline{x}^*(e) = \overline{x}(e) - \epsilon$ if $e = e_2$, where ϵ is a positive scalar sufficiently small. We then have that $\overline{x}^*$ is also a solution of $S(\overline{x})$, which is a contradiction. We claim that E_0 does not contain any edge with fractional value. Suppose, on the contrary that h is such an edge. Then $\overline{x}(E_0) > 1$. Therefore there exists a cut or a node-cut of system $S(\overline{x})$ containing h. Let v be an extremity of h. Let $\delta(S)$ be a cut of $\mathcal{C}^*_e(\overline{x})$ that contains h. Thus $E_0 \subset \delta(S)$. Suppose w.l.o.g., that $v \in \overline{S}$. Consider the node-cut $\delta_{G\setminus v}(S)$. We have that $\overline{x}(\delta_{G\setminus v}(S)) \leqslant \overline{x}(\delta(S) \setminus E_0) < k - 1$, a contradiction. Now consider a node-cut $\delta_{G\setminus Z}(T)$ of $\mathcal{C}^*_n(\overline{x})$ that contains h and hence E_0. As $\overline{x}(E_0) > 1$, one must have $|Z| < k - 1$. So suppose that $|Z| < k - 1$. Suppose w.l.o.g., that $v \in V \setminus (T \cup Z)$. Let $Z' = Z \cup \{v\}$. We have $\overline{x}(\delta_{G\setminus Z'}(T)) \leqslant \overline{x}(\delta_{G\setminus Z}(T)) - 1 - \overline{x}(h) = k - (|Z| + 1) - \overline{x}(h) < k - |Z'|$, a contradiction. Consequently, $\overline{x}(e) = 1$ for all $e \in E_0$. From the development above we also deduce that neither a cut of $\mathcal{C}^*_e(\overline{x})$ nor a node-cut of $\mathcal{C}^*_n(\overline{x})$ intersects E_0. Hence $\mathcal{C}^*_e(\overline{x}) \cup \mathcal{C}^*_n(\overline{x}) \subset \mathcal{C}(\overline{x}')$. Moreover, we have that $\overline{x}' \in P(G',k)$. Obviously, $\overline{x}'$ satisfies the trivial inequalities as well as the cut and node-cut inequalities that do not contain h. Let $\delta(W)$ be a cut that contains h. Suppose $v \in \overline{W}$. We have that $\overline{x}'(\delta(W)) = \overline{x}'(h) + \overline{x}'(\delta(W) \setminus \{h\}) = 1 + \overline{x}(\delta(W) \setminus E_0) = 1 + \overline{x}(\delta_{G\setminus v}(W)) \geqslant k$. Consider now a node-cut $\delta_{G\setminus Z}(T)$ containing h. If $|Z| = k - 1$, as $\overline{x}'(h) = 1$ and $h \in \delta_{G\setminus Z}(T)$, we have that $\overline{x}'(\delta_{G\setminus Z}(T)) \geqslant 1$. If $|Z| < k - 1$, then let $Z' = Z \cup \{v\}$. We have that $\overline{x}'(\delta_{G\setminus Z}(T)) \geqslant 1 + \overline{x}'(\delta_{G\setminus Z'}(T)) \geqslant 1 + k - |Z'| = 1 + k - |Z| - 1 = k - |Z|$. □

As we will see later, the reduction operations $\theta_1, \ldots, \theta_4$ can be used as a preprocessing for the separation procedures in our Branch-and-Cut algorithm.

7. Branch-and-Cut algorithm

In this section, we present a Branch-and-Cut algorithm for the kNCSP. The algorithm is based on the theoretical results presented in the previous sections. We will first present the general framework of the algorithm, then we will address the main issues of our algorithm, that are the separation procedures for the various inequalities we will use, and a primal heuristic.

We will consider a graph $G = (V, E)$ and a weight vector $c \in \mathbb{R}^E$ associated with the edges of G. We let $k \geqslant 1$ be the connectivity requirement.

7.1. General framework

To start the optimization we consider the following linear program consisting in the cut constraints induced by node sets $\{u\}$, for every $u \in V$ together with the trivial inequalities, that is

$$\min \sum_{e \in E} c(e)x(e)$$
$$x(\delta_G(u)) \geqslant k \qquad \text{for all } u \in V,$$
$$0 \leqslant x(e) \leqslant 1 \qquad \text{for all } e \in E.$$

If the optimal solution $\bar{y} \in \mathbb{R}^E$ of the above LP is feasible for the kNCSP, that is, it is integer and it satisfies all the cut and node-cut inequalities, then it is optimal for the problem. Usually, $\bar{y}$ is not feasible for the kNCSP. Thus, we need to generate further valid inequalities for the problem which are violated by $\bar{y}$. This is done by addressing the separation problem associated with the cut and node-cut inequalities, respectively, and the other families of inequalities we consider in our algorithm. Recall that the *separation problem* associated with $\bar{y}$ and a family of inequalities $\mathcal{F}$ is to say if $\bar{y}$ satisfies or not all the inequalities of $\mathcal{F}$, and if not, exhibit at least one inequality of $\mathcal{F}$ which is violated by $\bar{y}$. An algorithm solving a separation problem is called a *separation algorithm*. In our algorithm, we use the inequalities that we described in the previous sections and perform their separation in the following order

1. cut inequalities,
2. node-cut inequalities,
3. SP-node-partition inequalities,
4. F-node-partition inequalities,
5. node-partition inequalities.

We move to a class of inequalities when the separation algorithm for the previous class of inequalities has not found any violated inequality. We may add several inequalities at the same time in the Branch-and-Cut algorithm. Moreover, all the inequalities are global, that is they are valid for all the nodes of the Branch-and-Cut tree.

Remark that the separations are done on the graph obtained after repeated applications of the reduction operations $\theta_1, \ldots, \theta_4$ to the graph G and solution $\bar{y}$. If G' is the reduced graph and $\bar{y}'$ is the restriction of $\bar{y}$ to G', then by Lemmas 3–6, $\bar{y}'$ is an extreme point of $P(G', k)$ if $\bar{y}$ is an extreme point of $P(G, k)$. Moreover, we have the following results which are easily seen to be true.

Lemma 7. *Let $a'x \geqslant \alpha$ be an F-node-partition inequality (respectively node-partition inequality) valid for kNCSP(G'), induced by a partition $\pi' = (V'_0, V'_1, \ldots, V'_p), p \geqslant 2$ and an edge set F (respectively $\pi' = (V'_1, \ldots, V'_p), p \geqslant 3$) of $V' \setminus Z$, with $Z \subset V$. Let $\pi = (V_0, V_1, \ldots, V_p)$, $p \geqslant 2$ (respectively $\pi = (V_1, \ldots, V_p), p \geqslant 3$) be the partition of V obtained by expanding the elements of π'. Let $ax \geqslant \alpha$ be the 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 $ax \geqslant \alpha$ is valid for kNCSP(G). Moreover, if $a'x \geqslant \alpha$ is violated by $\bar{y}'$, then $ax \geqslant \alpha$ is violated by $\bar{y}$.

Lemma 8. *Let $a'x \geqslant \alpha$ be an SP-node-partition inequality valid for kNCSP(G'), induced by a partition $\pi' = (V'_1, \ldots, V'_p), p \geqslant 3$ of $V' \setminus Z$, with $Z \subset V$ such that $|Z| \leqslant k - 1$. Let $\pi = (V_1, \ldots, V_p), p \geqslant 3$ be the partition of $V \setminus Z$ obtained by expanding the subsets V'_i of π'. Let $ax \geqslant \alpha$ be the lifted SP-node-partition inequality obtained from $a'x \geqslant \alpha$ by application of the lifting procedure described in Section 3.3 for the edges of $E \setminus E'$. Then $ax \geqslant \alpha$ is violated by $\bar{y}$, if $a'x \geqslant \alpha$ is violated by $\bar{y}'$.*

Lemmas 7 and 8 show that the separation of F-node-partition, SP-node-partition and node-partition inequalities can be done in the reduced graph associated with any fractional solution of $P(G, k)$.

7.2. Separation algorithms

Now we describe the separation algorithms we have devised for the cut, node-cut, SP-node-partition, F-node-partition and node-partition inequalities.

We start by the separation of the cut inequalities (3). It is well known that the separation of the cut inequalities (3) reduces to computing a minimum weight cut in G with respect to weight vector $\bar{y}$. Indeed, there is a violated cut inequality (3) if and only if the minimum weight of a cut, w.r.t. weight vector $\bar{y}$, is $< k$. One can compute a minimum weight cut in polynomial time by using any minimum cut algorithm, and especially by using the Gomory-Hu algorithm Gomory and Hu (1961) which computes the so-called Gomory-Hu cut tree. This algorithm consists in $|V| - 1$ maximum flow computations.

Now we discuss the separation of the node-cut inequalities (4). In what follows, we show that these inequalities can be separated in polynomial time. In fact, Grötschel et al. (1995) present a separation algorithm for inequalities (4) based on a transformation of the graph G into a directed graph $\widetilde{G} = (\widetilde{V}, \widetilde{A})$. This transformation is presented as follows. For each node $u \in V$, we add in $\widetilde{V}$ two copies u^- and u^+ of u. The arc set is built in the following way. First, for each edge $uv \in E$, we add two arcs (v^+, u^-) and (u^+, v^-). Finally, for every node $u \in V$, we add an arc of the form (u^-, u^+). We also let $\widetilde{y} \in \mathbb{R}^{\widetilde{A}}$ be a weight vector given by

$$\widetilde{y}(a) = \begin{cases} \bar{y}(uv) & \text{for } a = (u^+, v^-) \text{ and } a = (v^+, u^-), \\ 1 & \text{if } a = (u^-, u^+) \text{ for all nodes } u \in V. \end{cases}$$

One can see that a cut $\delta(W)$ in G corresponds to a dicut which does not contain an arc of the form (u^-, u^+). Conversely, a dicut $\delta^{\pm}_{\widetilde{G}}(\widetilde{W})$ of $\widetilde{G}$ which does not contain any arc of the form (u^-, u^+) corresponds to a cut of G. Also, a node-cut $\delta_{G \setminus Z}(W)$ of G corresponds to a dicut of $\widetilde{G}$ which contains $|Z|$ arcs of the form (u^-, u^+). Conversely, a dicut of $\widetilde{G}$ which contains at least one arc of the form (u^-, u^+) corresponds to a node-cut of G. The corresponding node set Z is given by the nodes $u \in V$ such that $(u^-, u^+) \in \delta^{\pm}_{\widetilde{G}}(\widetilde{W})$, and the edges of $\delta_{G \setminus Z}(W)$ are given by the arcs of $\delta^{\pm}_{\widetilde{G}}(\widetilde{W})$ of the form (u^+, v^-) with $u^+ \in \widetilde{W}, v^- \in \widetilde{V} \setminus \widetilde{W}$.

Thus, cuts and node-cuts of G corresponds to dicuts of $\widetilde{G}$ which does not contain arcs of the form (u^-, u^+), and vice versa. Moreover, we have that

- if $\delta(W)$ in G and $\delta^{\pm}_{\widetilde{G}}(\widetilde{W})$ in $\widetilde{G}$ are corresponding cuts, then $\bar{y}(\delta(W)) = \widetilde{y}(\delta^{\pm}_{\widetilde{G}}(\widetilde{W}))$;
- if $\delta_{G \setminus Z}(W)$ in G and $\delta^{\pm}_{\widetilde{G}}(\widetilde{W})$ in $\widetilde{G}$ are corresponding cuts, then $\bar{y}(\delta_{G \setminus Z}(W)) + |Z| = \widetilde{y}(\delta^{\pm}_{\widetilde{G}}(\widetilde{W}))$.

Thus, there is a cut or node-cut inequality violated by $\bar{y}$ if and only if there exists a dicut $\delta^{\pm}_{\tilde{G}}(\widetilde{W})$ in $\tilde{G}$ whose weight with respect to $\tilde{y}$ is $< k$. Notice that, if we assume the cut inequalities to be all satisfied by $\bar{y}$, finding violated node-cut inequalities then reduces to compute a minimum weight cut in $\tilde{G}$ w.r.t. weight vector $\tilde{y}$.

Consequently, our separation algorithm for node-cut inequalities is as follows. First, we assume that the cut inequalities are all satisfied by $\bar{y}$. We build the graph $\tilde{G}$ and compute a minimum weight cut, say $\delta^{\pm}_{\tilde{G}}(\widetilde{W}^*)$, w.r.t. $\tilde{y}$. If $\tilde{y}(\delta^{\pm}_{\tilde{G}}(\widetilde{W}^*)) \geqslant k$, then every node-cut inequality is satisfied by $\bar{y}$, and the algorithm stops. If $\tilde{y}(\delta^{\pm}_{\tilde{G}}(\widetilde{W}^*)) < k$, then there is a violated node-cut inequality induced by a node-cut $\delta_{G\setminus Z}(W)$ with $Z \subseteq V, |Z| \leqslant k-1$, and $W \subseteq V \setminus Z$. The node sets Z and W are given by

$$Z = \{u \in V \text{ such that } (u^-, u^+) \in \delta^{\pm}_{\tilde{G}}(\widetilde{W}^*)\},$$

$$W = \{u \in V \text{ such that } u^-, u^+ \in \widetilde{W} \text{ or } u^+ \in \widetilde{W} \text{ and } u^- \in \tilde{V} \setminus \widetilde{W}\}.$$

Finally, computing a minimum weight cut in $\tilde{G}$ can be done in polynomial time by computing, for every pair of nodes $(s,t) \in V \times V$, with $s \neq t$, a maximum flow in $\tilde{G}$ from source node s^+ to destination t^-. This, hence, reduces our algorithm to $|V|(|V|-1)/2$ maximum flow computations in $\widetilde{W}$, which is polynomial.

Finally, we consider the separation problems for node-partition, SP-node-partition and F-node-partition inequalities. First notice that the separation problem of node-partition inequalities is NP-Hard even when $Z = \emptyset$. For our purpose, we consider these inequalities in the case where $Z = \emptyset$. Thus, the corresponding node-partition, SP-node-partition and F-node-partition inequalities also correspond to partition, SP-partition and F-partition inequalities, which are valid for the kECSP on G. Therefore, to separate these inequalities, we use the separation heuristics developed in Bendali et al. (2010) for these latter inequalities. These algorithms are applied on the graph G' and solution $\bar{y}'$ obtained by the application of the reduction operations to G and $\bar{y}$. As mentioned before, by Lemmas 7 and 8, any violated node-partition, SP-node-partition and F-node-partition inequality found in G' by the separation procedure is valid for kNCSP(G) and is also violated by $\bar{y}$.

7.3. Primal heuristic

Next, we discuss a primal heuristic for the problem. The aim of this heuristic is to produce, for a given instance, good upper bounds of the optimal solution of the problem. Such upper bounds are used by the Branch-and-Cut algorithm to prune unrelevant branches of the Branch-and-Cut tree. This also ensures that Branch-and-Cut algorithm produces a feasible solution, even if it reaches the maximum CPU time.

The primal heuristic we have developed for our purpose consider a fractional solution $\bar{y}$ obtained at the end of the cutting plane phase, at the root node of the Branch-and-Cut tree. The aim of the heuristic is to transform $\bar{y}$ into a feasible solution for the problem. To do this, we first build the graph $\overline{G} = (V, \overline{E})$ obtained by removing from G every edge $e \in E$ with $\bar{y}(e) = 0$. Then, we iteratively remove

Table 1
Results for SNDLIB instances with $k = 3$.

Instance	#EC	#NC	#SPC	#FNPC	#NPC	COpt	Gap1	NSub1	CPU1
atlanta_15	15	606	1	17	1	3265	0.01	3	0:00:01
geant_22	72	1990	19	28	6	375	1.07	60	0:00:26
france_25	80	7500	15	36	7	3254	0.08	37	0:00:32
norway_27	68	4448	10	55	5	5730	0.76	15	0:00:43
sun_27	42	2582	8	28	0	4771	0.04	7	0:00:31
india_35	62	2231	5	26	6	452	0.33	8	0:00:53
cost266_37	135	10,726	30	775	7	275	0.9	13	0:18:01
giul_39	62	2760	7	32	1	5878	0.03	5	0:02:19
pioro_40	11	2866	0	2	0	5637	0.00	1	0:00:09
germany_50	42	13,094	5	14	2	112	0.01	4	0:02:37
ta2_65	124	7597	10	106	4	5334	0.07	9	0:43:55

Table 2
Results for TSPLIB instances with $k = 3$.

Instance	#EC	#NC	#SPC	#FNPC	#NPC	COpt	Gap1	NSub1	CPU1
bays_29	74	3709	11	39	8	14,815	1.01	19	0:01:10
dantzig_42	137	9156	12	32	16	1232	0.03	42	0:14:14
att_48	138	13,995	14	47	10	17,527	0.02	48	0:42:11
eil_51	55	4680	7	30	1	745	0.02	4	0:06:41
berlin_52	133	9518	26	95	10	12,644	0.22	30	0:27:05
eil_76	80	15,321	8	84	4	947	0.11	8	0:48:05
gr_96	174	330	19	6	0	915	0.6	2	2:03:11
rat_99	112	294	9	19	0	2105	0.3	32	2:02:26
kroA_100	169	305	24	13	1	36,492	0.21	2	2:04:35
rd_100	186	303	21	3	0	13,391	0.13	21	2:01:03
kroB_100	145	300	12	52	1	37,341	1.6	12	2:03:58
lin_105	214	317	15	6	6	24,870	2.4	35	2:01:44
gr_120	90	332	10	0	0	11,562	0.6	2	2:26:57
bier_127	136	364	16	2	0	199,863	3.2	23	2:42:41
pr_124	179	403	12	0	0	99,696	0.29	3	2:28:01
ch_130	122	371	10	0	0	10,571	7.1	12	2:48:25
kroA_150	130	415	1	2	0	44,952	2.6	23	2:49:56
*u_159	112	429	3	7	0	71,772	*8.9*	59	5:00:00

from $\overline{G}$ all the edges uv such that u and v are both incident in $\overline{G}$ to at least $k+1$ edges. We denote by $\overline{G}' = (V, \overline{E}')$ the resulting graph, and by z the incidence vector of $\overline{G}'$. Next, we check if $\overline{G}'$ is k-node-connected. We do this by calling the separation algorithms for the cut and node-cut inequalities described in the previous section on z and $\overline{G}'$. If there is a cut (resp. node-cut) inequality induced by a violated by z, then $\overline{G}'$ is feasible for the kNCSP. We repeat this procedure until the graph $\overline{G}'$ is k-node-connected.

Finally, the algorithm computes and returns the weight of the graph $\overline{G}'$ obtained at the end of the previous step. The whole procedure is summarized by Algorithm 1 below.

```
Data: An undirected graph G = (N, E), an integer k, a fractional solution \overline{y}
Result: An Upper Bound UB for the kNCSP
begin
    Build the graph \overline{G} = (V, \overline{E}) by removing from G every edge e such that \overline{y}(e) = 0;
    /*Remove the edges uv such that |\delta_{\overline{G}}(u)| \geq k+1 and |\delta_{\overline{G}}(v)| \geq k+1*/
    foreach edge uv \in \overline{E} do
        if |\delta_{\overline{G}}(u)| \geq k+1 and |\delta_{\overline{G}}(v)| \geq k+1 then
            \overline{E} \leftarrow \overline{E} \setminus \{uv\};
    /*Check if the resulting graph is k-node-connected*/
    FeasibleSolutionFound \leftarrow False;
    repeat
        Let z be the incidence vector of \overline{E};
        Call the separation procedure for cut inequalities;
        if there is a cut inequality violated by z then
            Let \delta_{\overline{G}}(W) be the cut inducing the violated cut inequality;
            Choose an edge e \in \delta_G(W) \setminus \overline{E} with minimum weight;
            \overline{E} \leftarrow \overline{E} \cup \{e\};
        else
            Call the separation procedure for node-cut inequalities with solution z
            and graph \overline{G};
            if there is a node-cut inequality violated by z then
                Let \delta_{\overline{G} \setminus Z}(W) be the node-cut inducing the violated node-cut
                inequality;
                Choose an edge e \in \delta_{G \setminus Z}(W) \setminus \overline{E} with minimum weight;
                \overline{E} \leftarrow \overline{E} \cup \{e\};
            else
                FeasibleSolutionFound \leftarrow True;
    until FeasibleSolutionFound = True;
    UB \leftarrow 0;
    foreach edge e \in \overline{E} do
        UB \leftarrow UB + c(e);
    return UB;
end
```

cut $\delta_{\overline{G}'}(W)$ (resp. node-cut $\delta_{\overline{G}' \setminus Z}(W)$), which is violated by z, then we add in $\overline{G}'$ an edge $e \in \delta_G(W) \setminus \overline{E}'$ (resp. $e \in \delta_{G \setminus Z}(W) \setminus \overline{E}'$) whose weight $c(e)$ is minimum. If there is no cut and node-cut inequality

8. Computational results

Now we present the computational results we have obtained with our Branch-and-Cut algorithm for the kNCSP. The algorithm has been implemented in C++ using CPLEX 12.5 (IBM) and Concert Technology library. All the experiments have been done on a computer equiped with a 2.10 GHz x4 Intel Core(TM) i7-4600U processor and running under linux with 16 GB of RAM. We have set the maximum CPU time to five hours. We have tested our algorithm on several instances composed of graphs taken from SNDLIB (SNDLIB) and TSPLIB (TSPLIB). These are complete graphs where each node is given coordinates in the plane. The weight of each edge uv is the rounded euclidian distance between the vertices u and v. The graphs we have considered have up to 65 nodes for SNDLIB graphs and up to 150 nodes for TSPLIB graphs.

The tests have been performed for $k = 3, 4, 5$, and in all the experiments, we have used the reduction operations described in the previous sections, unless specified.

For each instance, we have run the algorithm three times. The first run (Run 1) is performed with all the inequalities presented before and the reduction operations included in the algorithm. The second run (Run 2) is performed without the reduction operations. The third run (Run 3) is performed with the reduction operations and with only the cut and node-cut inequalities. The results are given in Tables 2–8. Each instance is given by its name followed by the number of nodes of the graph. The other entries of the tables are:

Algorithm 1. Primal Heuristic Algorithm for the kNCSP.

#EC	the number of generated cut inequalities
#NC	the number of generated node-cut inequalities
#SPC	the number of generated SP-node-partition inequalities
#NFPC	the number of generated F-node-partition inequalities
#NPC	the number of generated node-partition inequalities
COpt	the value of the optimal solution
Gap1	the relative error between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree during Run 1
Gap2	the relative error between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree during Run 2
Gap3	the relative error between the best upper bound and the lower bound obtained at the root node of the Branch-and-Cut tree during Run 3
NSub1	the number of nodes in the Branch-and-Cut tree obtained at Run 1
NSub2	the number of nodes in the Branch-and-Cut tree obtained at Run 2
NSub3	the number of nodes in the Branch-and-Cut tree obtained at Run 3
CPU1	the total CPU time in hh:mn:sec achieved at Run 1
CPU2	the total CPU time in hh:mn:sec achived at Run 2
CPU3	the total CPU time in hh:mn:sec achived at Run 3

The gaps are all given in percentage. The instances indicated with "$*$" are those for which the maximum CPU time has been reached by the Branch-and-Cut algorithm.

We start our experiments by running the algorithm with $k = 3$, for both SNDLIB and TSPLIB graphs, and using all the inequalities and the reduction operations, that is Run 1. The results are given in Tables 1,2.

We first observe that all the SNDLIB instances of Table 1 have been solved to optimality within the CPU time limit. For TSPLIB graphs, all the instances have been solved to optimality, except one, u_159. The CPU time for the instances solved to optimality is less than 45 min for SNDLIB instances and less than 2h50min for TSPLIB instances. We also observe that the gap between the optimal solution and the lower bound achieved at the root node of the Branch-and-Cut tree is less than 1% for all the SNDLIB instances except one, geant_22, for which the gap is 1.07%. For TSPLIB graphs, the gap is less than 1%, except for 6 instances for which the gap is less than 7.1%. We can also notice that the number of nodes in the Branch-and-Cut tree is quite small, less than 60 nodes, for all the instances. Our separation procedures have also detected several inequalities of each type (cut, node-cut, F-node-partition, SP-node-partition and node-partition inequalities), especially the cut and node-cut inequalities. Moreover, a large number of F-node-partition and SP-node-partition inequalities are generated while few node-partition inequalities have been generated. From these observations, we conclude that our Branch-and-Cut algorithm is efficient for solving the kNCSP with $k = 3$.

We have also run the algorithm, during Run 1, with $k = 4$ and $k = 5$. The results for $k = 4$ are given by Table 3, for SNDLIB instances, and by Table 4 for TSPLIB instances. Note that when k is even, the SP-node-partition and partition inequalities we have considered in our algorithm are redundant with respect to the cut inequalities. Thus, they are not used in the Branch-and-Cut algorithm for $k = 4$ and do not appear in Tables 3, 4.

We can first observe that for $k = 4$ all the SNDLIB instances are solved to optimality, in less than 2 min, and this, at the root node of the Branch-and-Cut tree. For TSPLIB instances, the problem is solved in less than 2h30 for all the instances with few nodes (less than 23) in the Branch-and-Cut tree. For these latter instances, 11 of them over 17 instances are solved at the root node of the Branch-and-Cut tree. As for $k = 3$, several cut and node-cut inequalities have been generated, and few F-node-partition inequalities are generated. The comparison with $k = 3$ shows that the problem seems easier when $k = 4$, since the optimal solutions are obtained faster when $k = 4$ for all the instances. For example, ta_65 is solved in 43min55sec with 9 nodes in the Branch-and-Cut tree when $k = 3$, while it is solved in 2 min at the root node of the Branch-and-Cut tree when $k = 4$. Moreover, instance u_159 is solved to optimality in 4h38min14sec when $k = 4$, whereas it is not solved to optimality within 5 h when $k = 3$.

We have also run our algorithm for $k = 5$. The results are given in Tables 5, 6.

Here also, we can see that several instances are solved to optimality at the root node of the Branch-and-Cut tree for both SNDLIB and TSPLIB instances. The comparison with $k = 3$ also shows that the problem seems easier when $k = 5$. Indeed, for SNDLIB graphs, 8 instances over 11 have been solved at the root node of the Branch-and-Cut tree when $k = 5$ whereas only one instance has been solved at the root node for $k = 3$. The observation is the same for TSPLIB instances. Here, 5 instances have been solved at the root node when $k = 5$ whereas no instance has been solved at the root node when $k = 3$. Also the CPU time is, in general, better when $k = 5$. For example, instance gr_120 is solved in 2h26min57sec when $k = 3$ and in 1h09min03sec when $k = 5$. All these observations suggest that the kNCSP becomes easier when k increases.

A comparison between the case $k = 4$ and $k = 5$ shows that the problem seems easier when $k = 4$. Indeed, the CPU time is in general better when $k = 4$ and fewer nodes are generated in the Branch-and-Cut tree when $k = 4$. We can say from this that the problem is harder when k is odd than when k is even. The remarks made here are similar to those made by Bendali et al. (2010) for the kECSP. They also concluded from their experiments that the kECSP is harder when k is odd, and that the kECSP becomes easier when k increases with the same parity.

Table 3
Results for SNDLIB instances with $k = 4$.

Instance	#EC	#NC	#FNPC	COpt	Gap1	NSub1	CPU1
atlanta_15	0	246	2	4615	0.00	1	0:00:01
geant_22	0	912	0	521	0.00	1	0:00:01
france_25	0	594	0	4692	0.00	1	0:00:01
norway_27	0	793	4	8257	0.00	1	0:00:03
sun_27	0	696	0	6867	0.00	1	0:00:01
india_35	4	1324	2	640	0.00	1	0:00:08
cost266_37	0	1326	0	392	0.00	1	0:00:03
giul_39	0	1602	2	8314	0.00	1	0:00:07
pioro_40	0	1711	3	8137	0.00	1	0:00:15
germany_50	0	2610	0	156	0.00	1	0:00:12
ta2_65	0	4417	4	7631	0.00	1	0:02:01

Table 4
Results for TSPLIB instances with $k = 4$.

Instance	#EC	#NC	#FNPC	COpt	Gap1	NSub1	CPU1
bays_29	4	897	0	20,945	0.00	1	0:00:01
dantzig_42	10	1858	9	1776	0.00	1	0:00:11
att_48	20	2458	5	17,380	0.00	1	0:01:15
eil_51	0	2544	0	1051	0.00	1	0:00:12
berlin_52	6	2860	2	18,351	0.00	1	0:00:54
eil_76	0	5981	2	1350	0.00	1	0:03:24
gr_96	62	438	76	1314	1.6	6	2:00:03
rat_99	29	10,135	14	3045	0.00	1	0:49:33
kroA_100	22	15,223	6	53,111	0.00	1	0:32:25
rd_100	81	451	32	20,341	1.9	4	2:02:38
kroB_100	95	5012	31	55,182	2.5	8	0:45:09
lin_105	33	11,339	4	36,430	0.00	1	0:36:58
gr_120	6	14,765	6	18,714	0.00	1	0:48:40
pr_124	69	7553	16	144,715	3.5	2	2:12:59
bier_127	30	16,447	0	283,154	0.00	1	0:34:22
ch_130	26	532	10	15,123	2.3	23	2:06:45
kroA_150	19	595	4	68,281	5.3	22	2:24:45
u_159	13	630	6	104,664	7.2	15	4:38:14

Table 5
Results for SNDLIB instances with $k = 5$.

Instance	#EC	#NC	#SPC	#FNPC	#NPC	COpt	Gap1	NSub1	CPU1
atlanta_15	14	326	0	12	0	6239	0.00	1	0:00:01
geant_22	2	1193	0	0	2	717	0.35	12	0:00:04
france_25	25	832	0	40	1	6478	0.00	1	0:00:20
norway_27	27	944	0	47	0	11,217	0.00	1	0:00:51
india_35	30	2023	0	21	0	864	0.17	1	0:00:51
sun_27	27	1296	0	46	0	9383	0.34	4	0:00:32
cost266_37	37	1677	0	149	1	527	0.19	4	0:04:38
giul_39	39	1838	0	75	0	11,264	0.00	1	0:03:41
pioro_40	0	1728	0	4	0	10,952	0.00	1	0:00:12
germany_50	9	2651	0	4	0	206	0.00	1	0:00:31
ta2_65	67	4723	0	103	0	10,276	0.00	1	0:45:20

The next series of experiments concerns the efficiency of the reduction operations $\theta_1, \ldots, \theta_4$. For this, we have run the Branch-and-Cut algorithm with $k = 3$ and without the reduction operations (Run 2).The results are given by Table 7.

We can observe from Table 7 that, for the considered instances, the performances of the Branch-and-Cut algorithm are decreased when the reduction operations are not used. One can see that both the CPU time and the number of nodes in the Branch-and-Cut tree are increased when the reduction operations are not used in the algorithm. Also, the gap increases for all the instances, which indicates that a fewer number of inequalities or less efficient inequalities are generated during the separation phases. Moreover, one instance, eil_76 which is solved to optimality at Run 1 is not solved to optimality within 5 h without the reduction operations. This clearly proves the efficiency of the reduction operations on the resolution process.

Our last series of experiments aims to check the efficiency of the *F*-node-partition, SP-node-partition and node-partition inequalities in solving the *k*NCSP. For this, we have run the Branch-and-Cut algorithm in Run 3 with only the cut and node-cut inequalities. The results are presented in Table 8.

Here also, the comparison between Run 1 and Run 3 shows that the performances are decreased when *F*-node-partition, SP-node-partition and node-partition inequalities are not used in the

Table 6
Results for TSPLIB instances with $k = 5$.

Instance	#EC	#NC	#SPC	#FNPC	#NPC	COpt	Gap1	NSub1	CPU1
bays_29	30	1066	0	64	0	28,504	0.00	1	0:01:09
dantzig_42	25	1989	1	4	2	1931	0.00	1	0:00:36
att_48	59	2597	0	25	4	17,945	0.21	19	0:05:10
eil_51	51	3012	0	187	0	1435	0.21	6	0:27:27
berlin_52	28	5401	0	17	4	24,913	0.05	1	0:04:41
eil_76	0	6030	0	3	0	1792	0.00	1	0:04:27
gr_96	48	14,203	3	32	4	1792	0.16	6	1:38:50
rat_99	62	561	0	37	0	4113	1.2	4	2:01:15
kroA_100	61	10,496	0	82	4	72,119	0.14	9	2:03:22
rd_100	73	10,485	0	60	8	27,273	0.07	9	2:02:07
kroB_100	44	581	0	55	6	75,143	0.31	8	2:01:37
lin_105	35	588	0	16	0	50,669	0.6	12	2:00:50
gr_120	0	14,814	0	9	2	23,135	0.00	1	1:09:03
pr_124	45	629	0	15	2	199,713	3.4	2	2:03:39
bier_127	65	763	0	14	0	391,092	4.1	2	2:01:48
ch_130	8	595	0	12	0	21,618	3.8	2	2:09:41
kroA_150	13	681	0	8	0	88,237	2.6	25	2:22:03
*u_159	5	630	0	2	0	75,915	9.3	91	5:00:00

Table 7
Comparison of results for $k = 3$ with and without the reduction operations.

Instance	Gap1	Gap2	NSub1	NSub2	CPU1	CPU2
atlanta_15	0.01	0.02	3	15	00:00:01	00:00:32
geant_22	1.07	1.94	60	77	00:00:26	00:01:21
france_25	0.08	0.09	37	57	00:00:32	00:01:48
norway_27	0.76	1.78	15	34	00:00:43	00:02:05
india_35	0.33	2.05	8	24	00:00:53	00:02:08
giul_39	0.03	1.4	5	11	00:02:19	00:15:34
ta2_65	0.07	1.7	9	35	00:43:55	02:42:37
dantzig_42	0.03	0.79	42	74	00:14:14	01:37:37
att_48	0.02	2.4	48	68	00:42:11	02:39:38
eil_76	0.11	3.4	8	53	00:48:05	05:00:00
gr_96	0.6	7.8	2	41	02:03:11	03:24:57

Table 8
Comparison of results for $k = 3$ with and without the *F*-node-partition, SP-node-partition and node-partition inequalities.

Instance	Gap1	Gap3	NSub1	NSub3	CPU1	CPU3
atlanta_15	0.01	0.03	3	9	00:00:01	00:00:45
geant_22	1.07	2.1	60	84	00:00:26	00:02:09
france_25	0.08	0.17	37	43	00:00:32	00:02:33
norway_27	0.76	1.97	15	60	00:00:43	00:02:51
india_35	0.33	1.71	8	17	00:00:53	00:04:14
giul_39	0.03	1.8	5	14	00:02:19	00:17:57
ta2_65	0.07	1.2	9	21	00:43:55	01:18:37
dantzig_42	0.03	1.64	42	57	00:14:14	00:45:52
att_48	0.02	1.9	48	71	00:42:11	01:34:59
eil_76	0.11	6.5	8	64	00:48:05	02:54:35
gr_96	0.6	13.2	2	34	02:03:11	05:00:00

Table 9
Comparison between of the best solutions of *k*ECSP and the *k*NCSP for $k = 3$.

Instance	COpt_3ECSP	COpt_3NCSP	Gap
dantzig_42	1210	1232	1.82
att_48	17,499	17,527	0.16
berlin_52	12,601	12,644	0.34
eil_76	876	947	8.11
rat_99	2029	2105	3.75
kroA_100	36,337	36,492	0.43
kroB_100	37,179	37,341	0.44
rd_100	13,284	13,391	0.81
gr_120	11,442	11,562	1.05
bier_127	198,184	199,863	0.85
ch_130	10,400	10,571	1.64
kroA_150	44,718	44,952	0.52

algorithm. The CPU time, the number of nodes in the Branch-and-Cut tree and the gap are increased for all the instances of Table 8. Also, instance gr_96 is not solved to optimality within 5 h when *F*-node-partition, SP-node-partition and node-partition inequalities are not used, while it is in Run 1. This also shows the efficiency of the above inequalities in solving the *k*NCSP.

We conclude this computational study by comparing the optimal solutions obtained here for the *k*NCSP with those of the *k*ECSP obtained by Bendali et al. (2010). The aim is to know how often optimal solutions of the *k*ECSP and *k*NCSP are equal. The next table, Table 9, presents, for some TSPLIB instances, the optimal solutions of the *k*ECSP, those of the *k*NCSP for $k = 3$ and the gap between the two solutions, given by

$$Gap = \frac{COpt\_3NCSP - COpt\_3ECSP}{COpt\_3ECSP}.$$

From Table 9, we can see that the optimal solutions of the two problems are different for all the considered instances. However, we can see that the gap between the two solutions is relatively small for most of them. This let us conclude that the best solutions obtained by Bendali et al. (2010) for the 3ECSP are good upper bounds of the optimal solutions of the 3NCSP. Clearly, this remark cannot be generalized since we may find graphs for which the gap between the optimal solutions of the kNCSP and the kECSP is more important, but solving the kECSP could produce a good approximation of the kNCSP.

9. Conclusion

In this paper we have studied the k-node-connected subgraph problem with high connectivity requirement, that is, when $k \geqslant 3$. We have presented some classes of valid inequalities and described some conditions for these inequalities to be facet defining for the associated polytope. We have also investigated the structural properties of the extreme points of the linear relaxation of the problem and presented some reduction operations. Using these results, we have devised a Branch-and-Cut algorithm for the problem. The computational results we have obtained have shown that the F-node-partition, SP-node-partition and partition inequalities are effective for solving the problem. Also, the reduction operations we have used are shown to be efficient in the separation phase of the Branch-and-Cut algorithm. The experiments also show that, as for the kECSP, the kNCSP becomes easier when k increases, and is harder when k is odd than when k is even.

The study presented in this paper shows the efficiency of some valid inequalities, namely F-node-partition, SP-node-partition and partition inequalities, in solving the kNCSP. It would be interesting to investigate the polytope of the problem in a deeper way and identify cases in which these inequalities completely define the polytope of the problem.

Also, one can consider the k-node-connectivity in other survivable network design problems. For instance, one can consider the design of k-node-connected networks with hop-constraints, that is when the length of the paths between the nodes does not exceed a given positive integer L. The kNCSP with hop-constraints may be more challenging than the kNCSP itself. The investigations on this problem will be the subject of future works.

Acknowledgements

We would like to thank the anonymous referee for his/her valuable comments that permitted to improve the presentation of the paper.

References

Bendali, F., Diarrassouba, I., Didi Biha, M., Mahjoub, A. R., & Mailfert, J. (2010). A Branch-and-Cut algorithm for the k-edge-connected subgraph problem. *Networks, 55*, 13–32.
Chopra, S. (1994). The k-edge connected spanning subgraph polyhedron. *SIAM J Discrete Mathematics, 7*, 245–259.
Cornuéjols, G., Fonlupt, J., & Naddef, D. (1985). The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming, 33*, 1–27.
Diarrassouba, I., & Slama, L. (2007). Les inégalités de SP-partition pour le problème du sous-graphe k-arête-connexe. Research Report, LIMOS RR-07-13.
Diarrassouba, I., Kutucu, H., & Mahjoub, A. R. (2016). Two node-disjoint hop-constrained survivable network design and polyhedra. *Networks, 67*, 316–337.
Didi Biha, M., & Mahjoub, A. R. (1996). k-edge connected polyhedra on series-parallel graphs. *Operations Research Letters, 19*, 71–78.
Garey, M. R., & Johnson, D. J. (1979). *Computer and intractability: A guide to the theory of NP-completness*. San Francisco: Freeman.
Gomory, R. E., & Hu, T. C. (1961). Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics, 9*, 551–570.
Grötschel, M., & Monma, C. L. (1990). Integer polyhedra arising from certain network design problems with connectivity constraints. *SIAM Journal on Discrete Mathematics, 3*, 502–523.
Grötschel, M., Monma, C. L., & Stoer, M. (1991). Polyhedral approaches to network survivability. *Series in Discrete Mathematics and Theoretical Computer Science, 5*, 121–141.
Grötschel, M., Monma, C. L., & Stoer, M. (1992). Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research, 40*, 309–330.
Grötschel, M., Monma, C. L., & Stoer, M. (1995). Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Operations Research, 43*, 1012–1024.
IBM. IBM ILOG CPLEX Optimization Studio 12.5 Documentation. <http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.5.1/maps/ic-homepage.html>.
Kerivin, H., & Mahjoub, A. R. (2015). Design of survivable networks: A survey. *Networks, 46*, 1–21.
Mahjoub, A. R. (1994). Two-edge connected spanning subgraphs and polyhedral. *Mathematical Programming, 64*, 199–208.
Mahjoub, A. R., & Nocq, C. (1999). On the linear relaxation of the 2-node connected subgraph polytope. *Discrete Applied Mathematics, 95*(1–3), 389–416.
SNDLIB. <http://sndlib.zib.de/home.action>.
TSPLIB. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.

Further reading

Bendali, F., Diarrassouba, I., Mahjoub, A. R., & Mailfert, J. (2010). The k edge-disjoint 3-hop-constrained paths polytope. *Discrete Optimization, 7*, 222–233.
Didi Biha, M., & Mahjoub, A. R. (2004). The k-edge connected subgraph problem I: Polytopes and critical extreme points. *Linear Algebra and Its Applications, 381*, 117–139.
Fonlupt, J., & Mahjoub, A. R. (2006). Critical extreme points of the 2-edge connected spanning subgraph polytope. *Mathematical Programming, 105*, 289–310.
Fortz, B., Mahjoub, A. R., Thomas McCormick, S., & Pesneau, P. (2006). Two-edge connected subgraphs with bounded rings: Polyhedral results and Branch-and-Cut. *Mathematical Programming, 105*(1), 85–111.
Schrijver, A. (2003). Combinatorial optimization: Polyhedra and efficiency.