The k-node-connected subgraph problem: Facets and Branch-and-Cut

Ibrahima Diarrassouba
LMAH
Université du Havre
Le Havre, France
Email: diarrasi@univ-lehavre.fr

Meriem Mahjoub
LAMSADE
Université Paris Dauphine
Paris, France
Email: meriem.mahjoub@dauphine.fr

A. Ridha Mahjoub
LAMSADE
Université Paris Dauphine
Paris, France
Email: mahjoub@dauphine.fr

***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 and discuss some structure properties and reduction operations. Using these results, we devise a Branch-and-Cut algorithm along with 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 [14], [15], 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. 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* $(k \geq 0)$ 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 to the design of reliable communication and transportation networks ([1], [10], [11], [12], [13]). The kNCSP is NP-hard for $k \geq 2$ ([9]). The edge version of the problem has been widely studied in the literature ([1], [3], [4], [10], [11], [12], [13], [17]). However, the kNCSP has been particulary considered for $k = 2$ (see [6], [16]). A little attention has been given for the high connectivity case where $k \geq 3$. The kNCSP has been studied by Grötschel et al. ([10], [11], [12], [13]) within a more general survivability model. In [11] Grötschel et al. introduce the concept of connectivity types. With each node $s \in V$ of G it is associated a nonnegative integer r_s, called the type of s. A subgraph of G is said to be survivable if for each pair of distinct nodes $s, t \in V$, the subgraph contains at least $r_{st} = min\{r_s, r_t\}$ edge (node) disjoint (s, t)-paths. Grötschel et al. study the problem from a polyhedral point of vue, and propose cutting plane algorithms [11], [12], [13].

In [6], Diarrassouba et al. consider the 2NCSP with bounded lengths. Here it is supposed that each path does not exceed L edges for a fixed integer $L \geq 1$. They investigate the polyhedral structure of the polytope and propose a Branch-and-Cut algorithm. In [16], Mahjoub and Nocq study the linear relaxation of the 2NCSP(G).
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.

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 $\delta(G)$ 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 $W_1, ..., W_p \subset V$, $p \geq 2$, we denote by $\delta_G(W_1, ..., W_p)$ the set of edges of G between the sets W_1, ..., W_p.

2. Formulation

If x^F is the incidence vector of the edge set F of a k-node connected spanning subgraph of G, then x^F satisfies

978-1-4673-8496-4/16/$31.00

the following inequalities (see [10]):

$x(e) \geq 0$	for all	$e \in E,$	(1)
$x(e) \leq 1$	for all	$e \in E,$	(2)
$x(\delta_G(W)) \geq k$	for all	$\emptyset \neq W \subseteq V,$	(3)
$x(\delta_{G\setminus Z}(W)) \geq k - \lvert Z\rvert$	for all	$\emptyset \neq Z \subseteq V,$ $\lvert Z\rvert \leq k-1;\ \emptyset \neq W \subseteq V\setminus Z.$	(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. Constraints (3) and (4) are called *cut inequalities* and *node-cut inequalities*, respectively. The kNCSP is equivalent to the linear integer program

$$\min\{cx \mid x \text{ satisfies } (1)-(4), x \in \{0,1\}^E\}.$$

We will denote by kNCSP(G) the convex hull of all the integer solutions of (1)-(4). It can be shown that it suffices to suppose that $|Z| = k-1$ for inequalities (4). It can also be easily seen that if G is $(k+1)$-node connected then kNCSP(G) is full dimentional.

3. Valid inequalities

In this section, we describe some classes of valid inequalities for kNCSP(G). Given a partition $\pi = (V_1, ..., V_p)$, $p \geq 2$, we will denote by G_π the subgraph induced by π, that is, the graph obtained by contracting the sets V_i, $i = 1, ..., p$, that is identifying all the nodes of V_i and preserving the adjacencies. Note that $\delta_G(V_1, ..., V_p)$ is the set of edges of G_π.

3.1. F-node-partition inequalities

Theorem 1. Let $Z \subset V$ with $|Z| \leq k-1$. Consider a partition $\pi = (V_0, ..., V_p)$ of $V\setminus Z$, and $Z_i = \{z \in Z \mid \exists\ e \in \delta(\{z\}, V_i)\}$ for $i = 1, ..., p$, and let F be a subset of $\delta_{G\setminus Z}(V_0)$ such that $\sum_{i=0}^{p}(k - |Z_i|) - |F|$ is odd. Then the inequality

$$x(\delta_{G\setminus Z}(\pi\setminus F)) \geq \lceil \frac{\sum_{i=0}^{p}(k-|Z_i|) - |F|}{2} \rceil \tag{5}$$

is valid for kNCSP(G).

3.2. SP-partition inequalities

In [3], Chopra introduces a class of valid inequalities when the graph G is outerplanar and k is odd. In [3], Didi Biha and Mahjoub extend this result as follows. Consider a partition $\pi = (V_1, ..., V_p)$ of V. If G_π is series-parallel and k is odd, then the inequality

$$x(\delta_G(V_1, ..., V_p)) \geq \lceil \frac{k}{2} \rceil p - 1 \tag{6}$$

is valid for the kNCSP(G). These inequalities are called SP-partition inequalities.

3.3. Node-partition inequalities

In [10], Grötschel et al. introduce a class of valid inequalities for kNCSP(G) as follows. Consider a subset $Z \subset V$, such that $|Z| \leq k-1$, and let $V_1, ..., V_p$, $p \geq 2$ be a partition of $V\setminus Z$. They show that if $p(k - |Z|)$ is odd, then the following inequality is valid for kNCSP(G).

$$x(\delta_{G\setminus Z}(V_1, ..., V_p)) \geq \begin{cases} \lceil \frac{p(k-|Z|)}{2} \rceil & if \quad |Z| \leq k-2 \\ p-1 & if \quad |Z| = k-1. \end{cases} \tag{7}$$

4. Facial aspect

Theorem 2.
Let $G = (V, E)$ be a graph and an integer $k \geq 3$. Let $Z \subset V$ and $Z_i = \{z \in Z \mid \exists\ e \in \delta(\{z\}, V_i)\}$. Suppose $|Z_i| \leq \frac{2}{3}(k-1)$, and for all $z \in Z$, $|\delta_G(\{z\})| \geq k+1$. Let $\pi = (V_0, V_1, ..., V_p)$, with $l \geq 1$, be a partition of $V\setminus Z$, such that

i) $|V_i| = 1$ or $G[V_i]$ is $(k+1)$-node-connected, for $i = 0, ..., p$,
ii) $|V_i| = 1$ or for all $u \in V_i$, $|u, V\setminus V_i]| \leq 1$, for $i = 0, ..., p$,
iii) $|[V_i, V_{i+1}]| \geq 1$, for $i = 1, ..., p$,
iv) $|[V_0, V_i]| \geq k - |Z_i| - 1$, for $i = 1, ..., p$, (see Figure 1) for an illustration with $k = 4$ and $p = 5$)

Let F_i be an edge subset of $[V_0, V_i]$ such that $|F_i| = k - |Z_i| - 1$, $i = 1, ..., p$. Let $F = \bigcup_{i=1}^{p} F_i$. Then the F-node-partition inequality

$$x(\delta_{G\setminus Z}(\pi\setminus F)) \geq \lceil \frac{\sum_{i=0}^{p}(k-|Z_i|) - |F|}{2} \rceil \tag{8}$$

induced by π and F, defines a facet of kNCSP(G).

Proof.

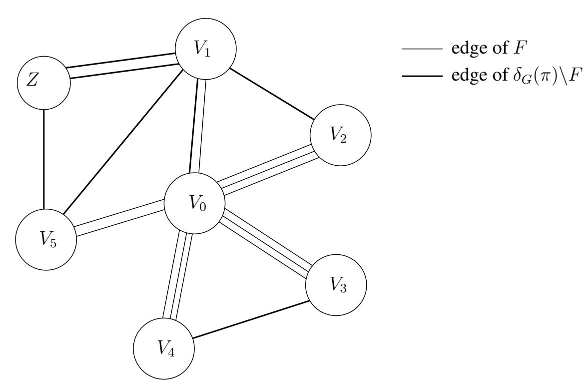

Figure 1. An F-node-partition configuration with $k = 4$

First observe that by Conditions i) - vi), G is (k+1)-node-connected and hence kNCSP(G) is full dimensional.

Let us denote the F-node-partition inequality by $ax \geq \alpha$ and let $\mathscr{F} = \{x \in kNCSP(G) | ax = \alpha\}$. Clearly, $\mathscr{F}$ is a proper face of kNCSP(G). Now suppose that there exists a defining facet inequality $bx \geq \beta$ such that $\mathscr{F} \subseteq F = \{x \in kNCSP(G) | bx = \beta\}$. We will show that $b = a$.

Let e_i be an edge of $[V_i, V_{i+1}]$, $i = 1, ..., p-1$. Let E_0 be the set of edges not in F and having both endnodes in the same element of π and $E_Z = E(Z) \cup \delta_G(Z)$. First we will show that $b(e) = 0$ for all $e \in E_0 \cup F \cup E_Z$. Let $i_0 \in \{1, ..., p\}$, and consider the edge sets

$$\begin{aligned} E_1 &= \{e_{i_0+2r}, r = 0, ..., \lfloor \frac{p-1}{2} \rfloor\} \\ T_1 &= E_0 \cup E_1 \cup F \cup E_Z \end{aligned}$$

Claim 1. T_1 induces a k-node-connected subgraph of G.

Proof. Let G_1 be the subgraph of G induced by T_1. Let $Z' \subset V$ with $|Z'| = k-1$. We will show that the graph $G_1 \backslash Z'$ is connected. Let $\delta_G(U)$ be a cut in $G_1 \backslash Z'$.

If $Z' \subset Z$. By Conditions i)-iv) $G_1 \backslash Z'$ is connected.

Now suppose that $Z' \subset V_i$, , $i = 1, ..., p$. By conditions i)-iv) we have that $G[V_i] \backslash Z'$ is connected and that there exist k-node-disjoint paths between a node in V_i and the rest of the graph. As $|Z'| = k-1$, in $G_1 \backslash Z'$ there exist at least one path connecting V_i to the rest of the graph. Hence $G_1 \backslash Z'$ is connected.

Now suppose that $Z' \subset V_0$. As $|Z_i| \leq \frac{2}{3}(k-1)$, $|Z'| \leq 2(k - |Z_i| - 1)$, and hence there exists at least one path between a node in V_i and V_0 in $G_1 \backslash Z'$. Hence $G_1 \backslash Z'$ is connected.

Suppose now that $Z' \subset (V_i \cup V_j)$, $i \neq j$. As $|Z'| \leq 2(k - |Z_i| - 1)$ and by Conditions i)-iv) we have that for all $u \in V_i$ there is at least one path between u and V_0. Hence $G_1 \backslash Z'$ is connected. Thus we have that $G_1 \backslash Z'$ is connected for every subset $Z' \subset V$ with $|Z'| = k-1$.

□

Note that there are $k+1$ node-disjoint paths connecting V_{i_0} to the rest of the graph induced by T_1. Now, observe that for any edge $e \in F_{i_0}$, one can show, in a similar way as in the claim above, that $T_2 = T_1 \backslash \{e\}$ also induces a k-node-connected subgraph of G. As x^{T_1} and x^{T_2} belong to $\mathscr{F}$, it follows that $bx^{T_1} = bx^{T_2} = \alpha$, implying that $b(e) = 0$ for all $e \in F_{i_0}$. As i_0 is arbitrarily chosen, we obtain that $b(e) = 0$ for all $e \in F$. Moreover, as the subgraphs induced by $V_0, ..., V_p$ are all $(k+1)$-node-connected, the subgraph induced by $T_1 \backslash \{e\}$, 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 $e \in E_Z$. As for every $z \in Z$ $|\delta_G(\{z\})| \geq k+1$, and by Condition ii) it follows that $T_1 \backslash \{e\}$ also induces a k-node-connected subgraph of G. Thus $b(e) = 0$ for all $e \in E_Z$. And consequently $b(e) = 0$ for all $e \in F \cup E_0 \cup E_Z$.

Next, we will show that $b(e) = a(e)$ for all $e \in \delta_{G \backslash Z'}(\pi) \backslash F$.

Consider the edge set $T_3 = (T_1 \backslash \{e_{i_0}\}) \cup \{e_{i_0+1}\}$.

We can show in a similar way as in the claim above that T_3 also induces a k-node-connected subgraph of G. Moreover, x^{T_3} belongs to $\mathscr{F}$, implying that $bx^{T_1} = bx^{T_3} = \alpha$. Hence $b(e_{i_0}) = b(e_{i_0+1})$. As e_{i_0} and e_{i_0+1} are arbitrary edges of $[V_{i_0}, V_{i_0+1}]$ and $[V_{i_0+1}, V_{i_0+2}]$, respectively, we obtain that $b(e)$ is the same for all $e \in [V_{i_0}, V_{i_0+1}] \cup [V_{i_0+1}, V_{i_0+2}]$. By exchanging the roles of V_{i_0},V_{i_0+1} and V_i, V_{i+1}, for $i = 1, ..., p-1$, we obtain by symmetry that $b(e) = \rho$ for all $e \in [V_i, V_{i+1}]$, $i = 1, ..., p$, for some $\rho \in \mathbb{R}$.

Let g_{i_0+1} be a fixed edge of $\delta_G(V_0) \backslash F$. Consider the edge set $T_4 = (T_1 \backslash \{e_{i_0}\}) \cup \{g_{i_0+1}\}$.

Similary, we can show that T_4 induces a k-node-connected subgraph of G. As x^{T_1} and x^{T_4} belong to $\mathscr{F}$, it follows in a similar way that $b(e_{i_0}) = b(g_{i_0+1})$. As $b(e_{i_0}) = b(e_{i_0+1}) = \rho$, we get $b(g_{i_0+1}) = \rho$. Here again, by exchanging the roles of V_{i_0+1} and V_i, $i = 1, ..., p$, we obtain that $b(e) = \rho$ for all $e \in [V_i, V_{i+1}] \cup \delta_G(V_0) \backslash F$, $i = 1, ..., p$. In consequence, the edges of $E \backslash (E_0 \cup F \cup E_Z)$ have all the same coefficient in $bx \geq \alpha$. Since $ax^{T_1} = bx^{T_1} = \alpha$, this yields $b(e) = 1$ for all $e \in E \backslash (E_0 \cup F \cup E_Z)$.

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

□

Corollary 1. If G is a complete graph, the F-node-partition inequalities defines facets only if $|V_i| = 1$, $i = 1, ..., p$.

5. Structural properties and reduction operations

In this section we discuss some structural properties of the extreme points of the linear relaxation of the problem and introduce some reduction operations with respect to extreme points.

Let $G = (V, E)$ be a graph. Let $P(G, k)$ be the polytope given by inequalities (1)-(4). Let $\overline{x}$ be an extreme point of $P(G, k)$. Let $\mathscr{C}_e(\overline{x})$ (resp. $\mathscr{C}_n(\overline{x})$) be the set of cuts $\delta(W)$ (resp. node-cuts $\delta_{G \backslash Z}(W)$) tight for $\overline{x}$, that is to say $\overline{x}(\delta(W)) = k$ (resp. $\overline{x}(\delta_{G \backslash Z}(W)) = k - |Z|$). Then $\overline{x}$ is the unique solution of a system of the form

$$\text{(Q)} \begin{cases} x(e) = 1, & \forall\ e \text{ such that } \ \overline{x}(e) = 1, \\ x(e) = 0, & \forall\ e \text{ such that } \ \overline{x}(e) = 0, \\ x(\delta_G(W)) = k, & \forall \ \ \delta_G(W) \in \mathscr{C}_e^*(\overline{x}), \\ x(\delta_{G \backslash Z}(W)) = k - |Z|, & \forall \ \ \delta_{G \backslash Z}(W) \in \mathscr{C}_n^*(\overline{x}), \end{cases}$$

where $\mathscr{C}_e^*(\overline{x})$ ($\mathscr{C}_n^*(\overline{x})$) is a subset of $\mathscr{C}_e(\overline{x})$ (resp. $\mathscr{C}_n(\overline{x})$).

Given a cut $\delta(W)$ (resp. a node-cut $\delta_{G \backslash Z}(W)$) tight for $\overline{x}$, let $\mathscr{C}_e(\overline{x}, W)$ (resp. $\mathscr{C}_n(\overline{x}, W)$) be the set of cuts $\delta(S)$ (node-cuts) $\delta_{G \backslash Z'}(T)$) tight for $\overline{x}$ such that either $S \subseteq W$ or $S \subseteq \overline{W}$ ($T \subseteq W$ or $T \subseteq \overline{W}$) (resp. $S \subseteq W$ or $S \subseteq V \backslash (W \cup Z)$ ($T \subseteq W$ or $T \subseteq V \backslash (W \cup Z)$)). Let $\mathscr{C}(\overline{x}, W) = \mathscr{C}_e(\overline{x}, W) \cup \mathscr{C}_n(\overline{x}, W)$.

Proposition 1. Let $\delta(W)$ ($\delta_{G \backslash Z}(W)$) be a cut (node-cut) of G tight for $\overline{x}$. Then system (Q) can be chosen so that $\mathscr{C}_e^*(\overline{x}) \cup \mathscr{C}_n^*(\overline{x})) \subseteq \mathscr{C}(\overline{x}, W)$.

Proof.

Let $\delta(W)$ be a cut of $\mathscr{C}_e(\overline{x})$. We can easily show that $\mathscr{C}_e^*(\overline{x})$ may be considered as a subset of $\mathscr{C}(\overline{x}, W)$. So consider a node-cut $\delta_{G\setminus Z}(T) \in \mathscr{C}_n(\overline{x})$. We distinguish two cases, either the subset Z is included in W or $\overline{W}$, or Z intersects W and $\overline{W}$. Consider the first case, and suppose W.l.o.g. that $Z \subset \overline{W}$. Also suppose that $T \cap W \neq \emptyset$ and $T \not\subset W$, $W \not\subset T$ and $T \cup W \neq V\setminus Z$. If this is not the case, then $\delta_{G\setminus Z}(T) \in \mathscr{C}(\overline{x}, W)$. Let $T_1 = T \cap W$, $T_2 = T \cap \overline{W}$, $T_3 = W\setminus T$ and $T_4 = \overline{W}\setminus(T \cup Z)$. Thus $T_i \neq \emptyset$ for $i = 1, ..., 4$. As $\delta(W) \in \mathscr{C}_e(\overline{x})$, we have that

$$\begin{aligned} k = \overline{x}(\delta(W)) = \overline{x}(\delta(T_1, T_2)) + \overline{x}(\delta(T_1, T_4)) \\ +\overline{x}(\delta(T_3, T_2)) + \overline{x}(\delta(T_3, T_4)) \\ +\overline{x}(\delta(T_1, Z)) + \overline{x}(\delta(T_3, Z)). \end{aligned} \tag{9}$$

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

$$\begin{aligned} k - |Z| = \overline{x}(\delta_{G\setminus Z}(T)) = \overline{x}(\delta(T_1, T_3)) + \overline{x}(\delta(T_1, T_4)) \\ +\overline{x}(\delta(T_2, S_3)) + \overline{x}(\delta(T_2, T_4)). \end{aligned} \tag{10}$$

Moreover, by considering the cuts $\delta(T_1)$, $\delta(T_3)$, and the node-cuts $\delta_{G\setminus Z}(T_2)$ and $\delta_{G\setminus Z}(T_4)$, we have that

$$\begin{aligned} k \leq \quad \overline{x}(\delta(T_1)) = \quad & \overline{x}(\delta(T_1, T_2)) + \overline{x}(\delta(T_1, T_3)) \\ & +\overline{x}(\delta(T_1, T_4)) + \overline{x}(\delta(T_1, Z)), \end{aligned} \tag{11}$$

$$\begin{aligned} k - |Z| \leq \quad & \overline{x}(\delta_{G\setminus Z}(T_4)) = \overline{x}(\delta(T_4, T_1)) \\ & +\overline{x}(\delta(T_4, T_2)) + \overline{x}(\delta(T_4, T_3)). \end{aligned} \tag{12}$$

As $\overline{x}(e) \geq 0$ for all $e \in E$, by (9) and (10) together with (11) and (12), it follows that

$$\overline{x}(\delta(T_2, T_3)) = 0, \qquad \overline{x}(\delta(T_3, Z)) = 0. \tag{13}$$

By symmetry we also get

$$\overline{x}(\delta(T_1, T_4)) = 0, \qquad \overline{x}(\delta(T_1, Z)) = 0. \tag{14}$$

In consequence, from (9), (10), (11), (12) together with (13) and (14), it follows that the cuts $\delta(T_1)$, $\delta(T_3)$ as well as the node-cuts $\delta_{G\setminus Z}(T_2)$ and $\delta_{G\setminus Z}(T_4)$ are all tight for $\overline{x}$.

Now observe that the equation $\overline{x}(\delta_{G\setminus Z}(T)) = k - |Z|$ is redundant with respect to the equations $\overline{x}(\delta(T_3)) = k$, $\overline{x}(\delta(T_1)) = k$, $\overline{x}(\delta_{G\setminus Z}(T_2)) = k - |Z|$, $\overline{x}(\delta_{G\setminus Z}(T_4)) = k - |Z|$ and the trivial equations. Moreover all these cuts are in $\mathscr{C}(\overline{x}, W)$.

Now suppose that $Z \cap W \neq \emptyset \neq Z \cap \overline{W}$. Also suppose that $T \cap W \neq \emptyset$ and $T \not\subset W$, $W \not\subset T$ and $T \cup W \neq V\setminus Z$. 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 $i = 1, ..., 4$. As $\delta(W) \in \mathscr{C}_e(\overline{x})$, we have that

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

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

$$\begin{aligned} k - |Z| = \overline{x}(\delta(T)) = \overline{x}(\delta(T_1, T_3)) + \overline{x}(\delta(T_1, T_4)) \\ +\overline{x}(\delta(T_2, T_3)) + \overline{x}(\delta(T_2, T_4)). \end{aligned} \tag{16}$$

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

$$\begin{aligned} k - |Z_1| \leq \overline{x}(\delta_{G\setminus Z_1}(T_1)) = \overline{x}(\delta(T_1, T_2)) \\ +\overline{x}(\delta(T_1, T_3)) + \overline{x}(\delta(T_1, T_4)) + \overline{x}(\delta(T_1, Z_2)), \end{aligned} \tag{17}$$

$$\begin{aligned} k - |Z_2| \leq \overline{x}(\delta(T_4)) = \overline{x}(\delta(T_4, T_1)) \\ +\overline{x}(\delta(T_4, T_2)) + \overline{x}(\delta(T_4, T_2)) + \overline{x}(\delta(T_4, Z_1)). \end{aligned} \tag{18}$$

As $\overline{x}(e) \geq 0$ for all $e \in E$, by (15) and (16) together with (17) and (18), it follows that

$$\begin{aligned} \overline{x}(\delta(T_2, T_3)) = 0, \qquad \overline{x}(\delta(T_3, Z_2)) = 0, \\ \overline{x}(\delta(T_2, Z_1)) = 0, \qquad \overline{x}(\delta(Z_1, Z_2)) = 0. \end{aligned} \tag{19}$$

By symmetry we also have that

$$\begin{aligned} \overline{x}(\delta(T_1, T_4)) = 0, \qquad \overline{x}(\delta(T_1, Z_2)) = 0, \\ \overline{x}(\delta(T_4, Z_1)) = 0. \end{aligned} \tag{20}$$

Now from (15), (16), (17), (18) together with (19), (20), it follows that the node-cuts $\delta_{G\setminus Z_1}(T_1)$ and $\delta_{G\setminus Z_2}(T_4)$ are tight for $\overline{x}$. Along the same line we obtain that the node-cuts $\delta_{G\setminus Z_1}(T_3)$ and $\delta_{G\setminus Z_2}(T_2)$ are also tight for $\overline{x}$. As a consequence we obtain that the equation $x(\delta_{G\setminus Z}(T)) = k - |Z|$ is redundant with respect to the equations $x(\delta_{G\setminus Z_1}(T_1)) = k - |Z_1|$, $x(\delta_{G\setminus Z_2}(T_4)) = k - |Z_2|$, $x(\delta(W)) = k$ together with the trivial equations $x(e) = 0$ for all e such that $\overline{x}(e) = 0$. Moreover all these cuts are in $\mathscr{C}(\overline{x}, W)$.

If we consider a node-cut $x(\delta_{G\setminus Z}(W)) \in \mathscr{C}_n^*(\overline{x}))$ we can show along the same line that the cuts of system (Q) can be chosen among those of $\mathscr{C}(\overline{x}, W)$.

□

In what follows we consider some reduction operations defined with respect to a solution $\overline{x}$ of $P(G, k)$.

- θ_1: Delete an edge $e \in E$ such that $\overline{x}(e) = 0$.
- θ_2: Contract a node subset $W \subseteq V$ such that $G[W]$ is k-edge connected, $\overline{x}(e) = 1$ for all $e \in E(W)$ and $\overline{x}(\delta(W)) = k$.
- θ_3: Contract a node subset $W \subseteq V$ such that $|W| \geq 2$, $|\overline{W}| \geq 2$, $|\delta_G(W)| = k$.
- θ_4: Replace a set of parallel edges by only one edge.
- θ_5: Contract a node subset W such that $\overline{x}(e) = 1$ for all $e \in E(W)$ and $|\delta_G(W)| \leq k + 1$.
- θ_6: Contract a node subset $T \cup Z$ such that $|\delta_{G\setminus Z}(T)| = k - |Z|$ and $\overline{x}(e) = 1$ for all $e \in E(T \cup Z)$.

Proposition 2.

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_2. Suppose that

1) $\overline{x}' \in P(G', k)$,

2) for all $Z \subset W$, $|Z| \leq k-1$, $\delta_{G\setminus Z}(T) \notin \mathscr{C}_n(\overline{x})$ for all $T \subseteq \overline{W}$.

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

Proof. As $\delta(W) \in \mathscr{C}_e(\overline{x})$, by Proposition 1, system (Q) can be chosen in such a way that for every $\delta(S) \in \mathscr{C}^*_e(\overline{x})$ (resp. $\delta_{G\setminus Z}(T) \in \mathscr{C}^*_n(\overline{x})$) either $S \subseteq W$ or $S \subseteq \overline{W}$ (resp. $T \subseteq W$ or $T \subseteq \overline{W}$). As $\overline{x}(e) = 1$ for all $e \in E(W)$ and $G[W]$ is k-edge connected, this implies that $\mathscr{C}^*_e(\overline{x}) \subseteq \mathscr{C}_e(\overline{x}')$. Moreover by 2) it follows that if $\delta_{G\setminus Z}(T)$ is tight for $\overline{x}$ and $Z \subseteq W$, then $W \cap T \neq \emptyset$ and $W\setminus(Z \cup T) \neq \emptyset$. Let $T_1 = W \cap T$ and $T_2 = W\setminus(Z \cup T)$. We have that

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

a contradiction. The last inequality comes from the fact that $G[W]$ is k-edge connected and $\overline{x}(e) = 1$ for all $e \in E(W)$. In consequence, all the node-cuts $\delta_{G\setminus Z}(T)$ of $\mathscr{C}^*_n(\overline{x})$ are such that $Z \subset \overline{W}$. However these are at the same time tight for $\overline{x}'$. Thus $\mathscr{C}^*_n(\overline{x}) \subset \mathscr{C}_n(\overline{x}')$. Let (Q') be the system obtained from (Q) by deleting the equations $x(e) = 1$ for all $e \in E(W)$. Then $\overline{x}'$ is the unique solution of (Q'). As all the equations of (Q') come from $P(G', k)$ and by 1) $\overline{x}' \in P(G', k)$, it follows that $\overline{x}'$ is an extreme point of $P(G', k)$.

□

Proposition 3.
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 (Q), 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 (Q) containing h. Let v be an extremity of h. Let $\delta(S)$ be a cut of $\mathscr{C}^*_e(\overline{x})$ 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 $\overline{x}(\delta_{G\setminus v}(S)) \leq \overline{x}(\delta(S)\setminus E_0) < k - 1$, a contradiction. Now consider a node-cut $\delta_{G\setminus Z}(T)$ of $\mathscr{C}^*_n(\overline{x})$ 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 $\overline{x}(\delta_{G\setminus Z'}(T)) \leq \overline{x}(\delta_{G\setminus Z}(T)) - 1 - \overline{x}(h) = k - (|Z| + 1) - \overline{x}(h) < k - |Z'|$, 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 $\mathscr{C}^*_e(\overline{x})$ nor a node-cut of $\mathscr{C}^*_n(\overline{x})$ intersects E_0. Hence $\mathscr{C}^*_e(\overline{x}) \cup \mathscr{C}^*_n(\overline{x}) \subset \mathscr{C}(\overline{x}')$. 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 $\overline{x}'(\delta(W)) = \overline{x}'(h) + \overline{x}'(\delta(W)\setminus\{h\}) = 1 + \overline{x}(\delta(W)\setminus E_0) = 1 + \overline{x}(\delta_{G\setminus v}(W)) \geq k$. 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 $\overline{x}'(\delta_{G\setminus Z}(T)) \geq 1$. If $|Z| < k - 1$, then let $Z' = Z \cup \{v\}$. We have that $\overline{x}'(\delta_{G\setminus Z}(T)) \geq 1 + \overline{x}'(\delta_{G\setminus Z'}(T)) \geq 1 + k - |Z'| = 1 + k - |Z| - 1 = k - |Z|$.

□

We can also show that the solution $\overline{x}'$ obtained by application of the other operations is an extreme point of $P(G', k)$ subject to some conditions, where G' is the graph that results from the operations.

We will use operations θ_1, ..., θ_6 as a preprocessing for the separation procedures in our Branch-and-Cut algorithm.

6. Branch-and-Cut algorithm

We now present our Branch-and-Cut algorithm for the kNCSP. The algorithm has been implemented in C++ using CPLEX 12.5 with the default settings. All experiments were run on a 2.10GHzx4 Intel Core(TM) i7-4600U running linux with 16 GB of RAM. We fixed the maximum CPU time to 5 hours. We have tested our approach on several instances derived from SNDlib[1] and TSPlib[2] based topologies. The test set consists in complete graphs whose edge weights are the rounded euclidian distance between the edge's vertices. The tests were performed for $k = 3, 4, 5$. In all our experiments, we have used the reduction operations described above. To start the optimization we consider the following linear program

$$\begin{aligned} &\min \sum_{e \in E} c(e)x(e) && \\ &x(\delta_G(u)) \geq k && \text{for all } u \in V, \\ &x(\delta_{G\setminus Z}(u)) \geq 1 && \text{for all } u \in V; Z \subseteq V; |Z| = k-1, \\ &0 \leq x(e) \leq 1 && \text{for all } e \in E. \end{aligned}$$

The inequalities previously described are separated in the following order: cut inequalities (3), node-cut inequalities (4), SP-partition inequalities (6), F-partition inequalities (8) and node-partition inequalities (7). The experimental results are summerized in the following tables.

1. http://sndlib.zib.de/home.action.
2. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html

978-1-4673-8496-4/16/$31.00 ©2016 IEEE

Instance	#EC	#NC	#FNPC	#SPC	#NPC	COpt	Gap(%)	NSub	CPU
schema1_8	0	284	0	1	2	146	0.00	1	0:00:01
dfn_bwin_10	0	0	0	0	0	44	0.00	1	0:00:01
di-yuan_11	4	92	4	3	1	2731	0.00	1	0:00:01
dfn_gwin_12	17	693	6	6	3	47	0.00	1	0:00:05
polska_12	7	121	0	5	0	51	0.00	1	0:00:04
abilene_12	20	1699	3	8	3	214	0.00	1	0:00:06
burma_14	19	3332	8	3	1	62	0.03	5	0:00:30
nobel-us_14	11	837	6	6	9	219	0.05	13	0:00:28
atlanta_15	23	34	10	2	3	3265	0.00	1	0:00:30
newyork_16	19	58	3	4	4	3809	0.00	1	0:00:39
ulysses16_16	22	5844	24	8	5	132	0.00	1	0:02:20
nobel_germany_17	40	1283	5	8	4	53	0.00	1	0:01:46
geant_22	78	2851	36	16	5	375	0.14	59	0:53:49
ulysses22_22	44	22444	8	7	9	141	0.00	1	0:13:33
ta1_24	46	520	11	7	2	3035	0.02	8	0:27:35
france_25	71	807	8	14	1	3254	0.39	17	1:03:55
janos-us_26	60	29258	42	13	3	282	0.01	12	1:40:27
sun_27	51	936	15	6	0	4771	0.02	8	1:12:58
norway_27	48	1214	52	7	6	6864	2.32	2	2:02:07
bays_29	66	227	22	8	6	14791	3.1	6	2:03:52
india_35	18	270	9	3	0	489	1.45	2	2:04:21
pioro_40	11	546	0	2	1	5637	0.00	1	0:42:07
berlin_52	95	914	14	83	0	16524	0.09	6	3:14:23
eil_76	85	1674	9	142	1	-	*0.12*	6	5:00:00

TABLE 1. RESULTS FOR $k = 3$

Instance	#EC	#NC	#FNPC	#SPC	#NPC	COpt	Gap(%)	NSub	CPU
schema1_8	0	0	0	-	0	207	0.00	1	0:00:01
dfn_bwin_10	0	0	0	-	0	44	0.00	1	0:00:01
di-yuan_11	4	92	4	-	1	2731	0.00	1	0:00:01
dfn_gwin_12	0	116	2	-	0	65	0.00	1	0:00:02
polska_12	0	1276	14	-	0	72	0.00	1	0:00:02
abilene_12	0	1252	6	-	2	305	0.00	1	0:00:03
burma_14	0	3966	3	-	0	85	0.00	1	0:00:12
nobel-us_14	0	6168	9	-	0	288	0.00	1	0:00:14
atlanta_15	0	72	2	-	0	4615	0.00	1	0:00:25
newyork_16	0	60	4	-	0	5462	0.00	1	0:00:20
ulysses16_16	0	45865	0	-	3	185	0.05	4	0:01:39
nobel_germany_17	20	4632	10	-	0	73	0.00	1	0:03:51
geant_22	4	118126	0	-	0	521	0.00	1	0:11:50
ulysses22_22	0	185103	0	-	0	196	0.00	1	0:09:46
ta1_24	18	5986	8	-	0	4387	0.00	1	0:37:00
france_25	4	0	1	-	0	4692	0.39	14	0:06:44
janos-us_26	24	15859	12	-	0	390	0.00	1	1:22:40
sun_27	0	9172	74	-	0	6867	0.00	1	0:21:31
norway_27	12	25377	6	-	5	8257	0.00	1	1:36:20
bays_29	4	12178	36	-	0	20945	0.00	1	1:00:38
india_35	25	4165	7	-	0	547	6.45	2	2:01:59
pioro_40	18	598	4	-	0	8096	0.00	1	0:31:14
berlin_52	145	1045	19	-	0	18268	0.05	3	3:14:23
eil_76	34	1832	8	-	0	971	0.07	2	2:24:32

TABLE 2. RESULTS FOR $k = 4$

Instance	#EC	#NC	#FNPC	#SPC	#NPC	COpt	Gap(%)	NSub	CPU
schema1_8	0	576	0	0	0	284	0.00	1	0:00:01
dfn_bwin_10	0	2600	0	0	0	81	0.00	1	0:00:01
dfn_gwin_12	9	5129	11	0	0	88	0.00	1	0:00:19
polska_12	0	15456	0	1	0	96	0.02	3	0:00:21
abilene_12	0	21027	0	0	2	437	0.05	6	0:00:39
burma_14	6	41956	2	0	2	111	0.01	3	0:01:52
nobel-us_14	13	50884	1	1	3	409	0.12	5	0:02:59
atlanta_15	15	32080	20	0	1	6239	0.00	1	0:13:44
newyork_16	0	44184	0	0	0	7422	0.00	1	0:01:31
ulysses16_16	0	133950	2	2	4	244	0.00	1	0:08:48
nobel_germany_17	16	66232	68	0	0	100	0.12	5	1:50:34
ulysses22_22	2	186516	0	1	2	258	0.02	2	2:07:16
ta1_24	0	435920	0	0	0	5915	0.00	1	0:47:17
france_25	0	21284	0	0	0	6439	0.91	5	2:23:00
janos-us_26	1	786529	0	0	0	52	5.5	6	2:40:41
sun_27	0	29916	1	0	0	9341	0.06	7	2:23:55
norway_27	0	29946	1	0	0	11149	0.65	7	2:23:55
bays_29	2	40972	0	0	0	28411	1.3	2	2:12:12
india_35	0	41344	0	0	0	638	2.91	4	1:46:04
pioro_40	8	1342	7	0	0	11756	0.25	4	0:58:24
berlin_52	76	3451	25	0	0	21763	0.15	5	4:14:23
st_70	4	847	21	0	0	-	9.12	1	5:00:00

TABLE 3. RESULTS FOR $k = 5$

Each instance is given by its name followed by an extension representing the number of nodes of the graph. The other entries of the table are: The connectivity (k), the number of generated cuts, for inequalities (3) (#EC) and (4) (#NC), respectively, the number of generated F-node-partition inequalities (8) (#FNPC), the number of generated SP-partition inequalities (6) (#SPC), the number of generated node-partition inequalities (7) (#NPC), the weight of the optimal solution obtained (COpt), the Gap, that is the relative error between the best upper bound (the optimal solution if the problem has been solved to optimality) and the lower bound obtained at the root node of the Branch-and-Cut tree, without using the additional valid inequalities (Gap_1), and by using them (Gap_2), the number of subproblems in the Branch-and-Cut tree (NSub), the total CPU time in h:min:sec, without using the valid inequalities (CPU_1), and by using them (CPU_2).

We have tested our Branch-and-Cut algorithm for differents connectiviy types. We first considered the case where $k = 3$. The results are summarized in Table 1. We can see from Table 1 that our Branch-and-Cut solved all the instances to optimality within the time limit of 5 hours except the last one. Moreover most of the instances have been solved in the cutting plane phase. We also notice that the relative error between the lower bound at the root node of the Branch-and-Cut tree and the best upper bound (Gap) is less than 1% for most of the instances. We also observe that our separation procedures detected an important number of violated SP-partition and specially F-partition inequalities, which are very efficient in the resolution of the problem.

Our second series of experiments concerns the kNCSP with $k = 4, 5$. The results are given in Table 2 for $k = 4$ and Table 3 for $k = 5$. When k is even, the SP-partition inequalities are redundant with respect to the cut inequalities (3). Thus we don't consider these inequalities in the resolution process for $k = 4$, and therefore they do not appear in Table 2.

First we can see that for $k = 4$, the CPU time is smaller than the one when $k = 3$. Moreover 19 instances over 24 have been solved in the cutting plane phase. A few number of violated node-partition inequalities are detected. However a large number of F-partition inequalities is generated. Thus these inequalities are very efficient for solving the kNCSP when k is even. Thus it appears that the kNCSP is easier to solve when k is even, and this is also confirmed by the results of Table 3. We can remark that the CPU time for all the instances when $k = 5$ is higher than that when $k = 4$. For instance, the test problem france_25 has been solved in 2h 23mn when $k = 5$, whereas only 6 minutes were needed to solve it for $k = 4$.

Instance	Gap1(%)	Gap2(%)	Gap_ ECSP(%)	CPU_1	CPU_2	CPU_ ECSP
schema1_8	0.02	0.00	-	0:00:06	0:00:01	-
polska_12	0.05	0.00	-	0:00:28	0:00:04	-
burma_14	0.12	0.03	0.00	0:01:54	0:00:30	0:00:01
nobel_germany_17	2.37	0.00	-	0:08:01	0:01:46	-
geant_22	4.14	0.14	-	0:01:11	0:53:49	-
ta1_24	3.21	0.02	0.01	1:01:51	0:27:35	0:00:01
france_25	2.94	0.39	0.02	1:51:46	1:03:55	0:00:02
janos-us_26	1.83	0.01	-	2:02:08	1:40:27	-
norway_27	6.51	2.32	-	2:44:36	2:02:07	-
india_35	5.43	1.45	0.13	3:33:12	2:04:21	0:00:02
pioro_40	3.43	0.00	-	2:42:51	0:42:07	-
berlin_52	*4.76*	0.09	0.45	5:00:00	3:14:23	0:00:03
eil_76	*5.92*	*0.12*	0.06	5:00:00	5:00:00	0:00:03

TABLE 4. COMPARISON OF RESULTS FOR $k = 3$

Figures 2, 3 and 4 give the optimal solutions of the instance "france_25" when $k = 3, 4, 5$, respectively.

To evaluate the impact of the F-node-partition inequalities and the other additional inequalities, we tried to solve the kNCSP by only separating the basic inequalities. Figure

Instance	Gap1(%)	Gap2(%)	Gap_ ECSP(%)	CPU_1	CPU_2	CPU_ ECSP
schema1_8	0.03	0.00	-	0:00:04	0:00:01	-
polska_12	0.01	0.00	-	0:00:04	0:00:02	-
burma_14	0.15	0.00	0.00	0:00:28	0:00:12	0:00:01
nobel_germany_17	0.62	0.00	-	0:04:21	0:03:51	-
geant_22	0.76	0.00	-	0:15:31	0:11:50	-
ta1_24	0.93	0.00	-	0:45:41	0:37:00	-
france_25	0.89	0.39	0.00	0:26:17	0:06:44	0:00:02
janos-us_26	3.62	0.00	0.00	2:00:01	1:22:40	0:00:02
norway_27	1.09	0.00	0.00	2:03:57	1:36:20	0:00:01
india_35	13.64	6.45	0.00	2:45:19	2:01:59	0:00:01
pioro_40	4.98	0.00	0.00	2:00:01	0:31:14	0:00:03
berlin_52	9.36	0.05	0.00	4:23:49	3:14:23	0:00:01
eil_76	*6.48*	0.07	-	5:00:00	2:24:32	-

TABLE 5. COMPARISON OF RESULTS FOR $k = 4$

Instance	Gap1(%)	Gap2(%)	Gap_ ECSP(%)	CPU_1	CPU_2	CPU_ ECSP
schema1_8	0.02	0.00	-	0:00:03	0:00:01	-
polska_12	0.03	0.02	-	0:00:58	0:00:21	-
burma_14	0.02	0.01	0.00	0:03:22	0:01:52	0:00:01
nobel_germany_17	0.34	0.12	-	2:03:37	1:50:34	-
ta1_24	0.27	0.00	-	0:59:57	0:47:17	-
france_25	1.22	0.91	0.00	2:31:21	2:23:00	0:00:01
janos-us_26	6.02	5.5	-	2:58:38	2:40:41	-
sun_27	0.92	0.06	-	3:52:14	2:23:55	-
norway_27	2.68	0.65	0.00	2:54:37	2:23:55	0:00:01
india_35	4.82	2.91	0.00	2:37:19	1:46:04	0:00:01
pioro_40	3.29	0.25	-	2:04:37	0:58:24	-
berlin_52	*2.62*	0.15	0.00	5:00:00	4:14:23	0:00:01

TABLE 6. COMPARISON OF RESULTS FOR $k = 5$

5 presents a fractional solution obtained at the root node using additional valid inequalities, whereas Figure 6 gives the solutions without these inequalities. We can see that without the valid inequalities, we have more edges with fractional values. Thus without additional constraints, the problem needs more branching. Moreover, the computional time is higher as it can be seen in Tables 4, 5 and 6. CPU_1 is more important that CPU_2, and Gap_1 is higher than Gap_2.

Instance	#EC	#NC	#FNPC	#SPC	#NPC	COpt	Gap(%)	NSub	CPU
abilene_12	20	2211	0	3	2	214	1.29	12	0:00:18
nobel-us_14	29	775	4	7	7	219	0.57	44	0:01:15
atlanta_15	16	143	16	1	1	3265	0.15	3	0:01:03
nobel_germany_17	83	5607	217	24	17	53	1.3	21	0:14:52
ulysses22_22	49	22633	5	10	15	141	0.74	10	0:21:08
janos-us_26	42	35782	26	2	0	282	1.45	26	3:14:53
sun_27	41	910	29	8	0	4771	0.87	6	3:31:11
norway_27	48	1214	52	7	6	6864	2.32	2	2:02:07
bays_29	66	227	22	8	6	14791	3.1	6	2:03:52
india_35	18	270	9	3	0	489	1.45	2	2:04:21
pioro_40	33	725	1	5	2	-	7.25	14	5:00:00

TABLE 7. RESULTS FOR $k = 3$ WITHOUT REDUCTION OPERATIONS

We also evaluated the impact of the reduction operations θ_1, ..., θ_6 on the separation procedures. We tried to solve the kNCSP, for $k = 3$, without using these operations. The results are given in Table 7. Observe that the CPU time increased for most of the instances. For instance, without the reduction operations, the instance pioro_40 has been solved to optimality after 5 hours. Whereas with the operations, it has been solved in 42mn 07s. Also, the CPU time for the instances janos-us_26 and sun_27 increased from 1 hour to more than 3 hours. Moreover, we remark that when we use the reduction operations, we generate more SP-partition, F-partition and node-partition inequalities and less nodes in the Branch-and-Cut tree than when we use them. This proves that our heuristics, used to separate the valid inequalities, are less efficient without the reduction operations. It then appears that the reduction operations play an important role in the resolution of the problem. They permit to much accelerate its resolution.

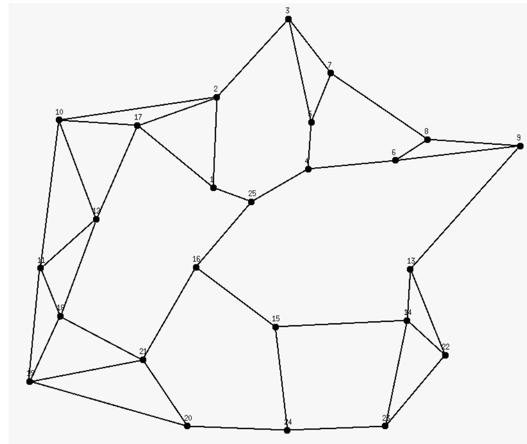

Figure 2. Solution of the kNCSP for $k = 3$

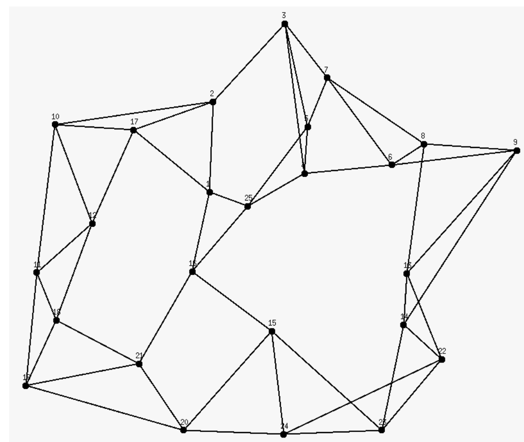

Figure 3. Solution of the kNCSP for $k = 4$

We also compared the results of the kNCSP with those of the k-Edge Connected Subgraph Problem (kECSP). Both problems are easier to solve when k is even. However, although the kECSP is easier to solve when k increases with the same parity, the kNCSP is not. This can be explaned by the fact that our separation procedure for the additional node-cut inequalities requires $C_{|V|}^{k-1}|V| - 1$ maximum flow computations, it will then take more time to run through the combination of $k - 1$ nodes of the graph when k increases.

7. Conclusion

In this paper we have studied the k-node-connected subgraph problem with high connectivity requirement, that is, when $k \geq 3$. We have presented some classes of valid inequalities and described some conditions for these inequalities to be facet defining for the associated polytope. Using this, we devised a Branch- and-Cut algorithm for the problem. This algorithm uses some reduction operations, and has been tested on SNDlib and TSPlib based instances.

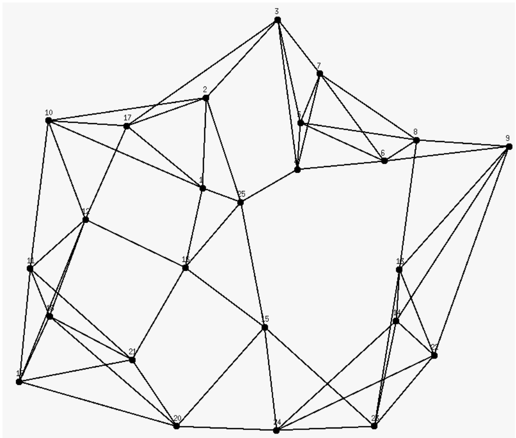

Figure 4. Solution of the kNCSP for $k = 5$

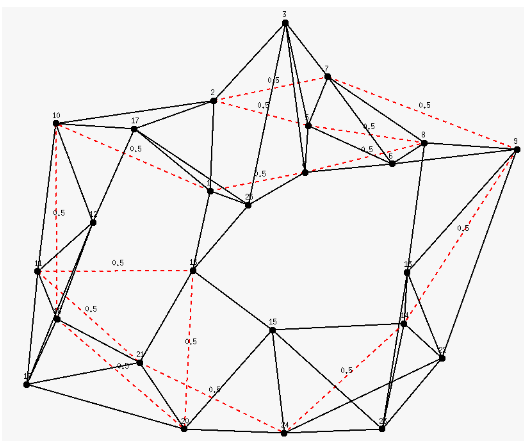

Figure 5. Fractional solution with valid inequalities

For future work, we can more investigate the structural properties of the linear relaxation and study the problem when a bound is considered on the connectivity paths.

References

[1] F. Bendali, I. Diarrassouba, M. Didi Biha, A.R. Mahjoub and J. Mailfert, *A Branch-and-Cut algorithm for the k-edge-connected subgraph problem*, Networks **55** (2010), 13-32.

[2] F. Bendali, I. Diarrassouba, A.R. Mahjoub, J. Mailfert, *The k edge-disjoint 3-hop-constrained paths polytope*, Discrete Optimization **7** (2010), 222-233.

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

[4] M. Didi Biha and A.R. Mahjoub, *k-edge connected polyhedra on series-parallel graphs*, Operations Research Letters **19** (1996), 71-78.

[5] M. Didi Biha and A.R. Mahjoub, *The k-edge connected subgraph problem I: Polytopes and critical extreme points*, Linear Algebra and its Applications **381** (2004), 117139.

[6] I. Diarrassouba, H. Kutucu and A. R. Mahjoub, *Two Node-Disjoint Hop-Constrained Survivable Network Design and Polyhedra*, ENDM **41**, (2013) 551-558.

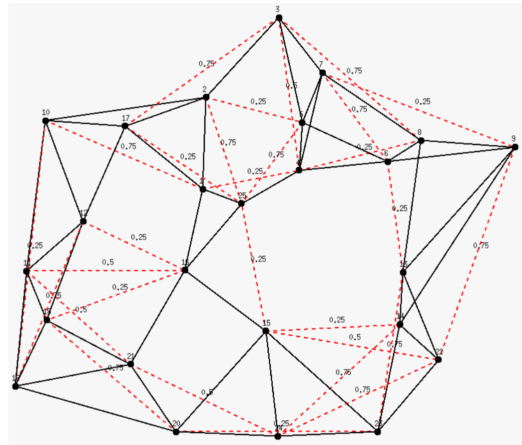

Figure 6. Fractional solution without valid inequalities

[7] J. Fonlupt and A. R. Mahjoub, *Critical extreme points of the 2-edge connected spanning subgraph polytope*, Mathematical Programming **105**, (2006) 289-310.

[8] B. Fortz, A. R. Mahjoub, S. Thomas McCormick and P. Pesneau, *Two-edge connected subgraphs with bounded rings: Polyhedral results and Branch-and-Cut*, Math. Program. 105 **1**, (2006) 85-111.

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

[10] M. Grötschel, C.L. Monma and M. Stoer, *Polyhedral Approaches to Network Survivability*, Series in Discrete Mathematics & Theoretical Computer Science **5** (1991), 121-141.

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

[12] M. Grötschel, C.L. Monma and M. Stoer, *Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints*, Operations Research **40** (1992), 309-330.

[13] M. Grötschel, C.L. Monma and M. Stoer, *Polyhedral and computational investigations for designing communication networks with high survivability requirements*, Operations Research **43** (1995), 1012-1024.

[14] H. Kerivin and A.R. Mahjoub, *Design of survivable networks: A survey*, Networks **46** (2015), 1-21.

[15] C.-W. Ko and C.L.Monma, *Heuristics for designing highly survivable sommunication networks*, (1989), New Jersey.

[16] A. R. Mahjoub, Charles Nocq, *On the Linear Relaxation of the 2-node Connected Subgraph Polytope*, Discrete Applied Mathematics 95 **1-3** (1999), 389-416.

[17] A. R. Mahjoub, Two-edge Connected spanning subgraphs and polyhedral, Mathematical Programmig, **64**, pp 199-208.