Annals of Telecommunications (2018) 73:5–28
https://doi.org/10.1007/s12243-017-0622-3

k-node-disjoint hop-constrained survivable networks: polyhedral analysis and branch and cut

Ibrahima Diarrassouba[1] · Meriem Mahjoub[2,3] · A. Ridha Mahjoub[2] · Hande Yaman[4]

Received: 19 October 2016 / Accepted: 17 December 2017 / Published online: 16 January 2018

Abstract
Given a graph with weights on the edges, a set of origin and destination pairs of nodes, and two integers $L \geq 2$ and $k \geq 2$, the k-node-disjoint hop-constrained network design problem is to find a minimum weight subgraph of G such that between every origin and destination there exist at least k node-disjoint paths of length at most L. In this paper, we consider this problem from a polyhedral point of view. We propose an integer linear programming formulation for the problem for $L \in \{2, 3\}$ and arbitrary k, and investigate the associated polytope. We introduce new valid inequalities for the problem for $L \in \{2, 3, 4\}$, and give necessary and sufficient conditions for these inequalities to be facet defining. We also devise separation algorithms for these inequalities. Using these results, we propose a branch-and-cut algorithm for solving the problem for both $L = 3$ and $L = 4$ along with some computational results.

Keywords k-node-disjoint hop-constrained paths · Survivable network · Polytope · Valid inequalities · Facets · Separation · Branch-and-cut

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 type of each telecommunication network. It is common to require several disjoint paths to link each pair of nodes to ensure the transmission in case of disconnection or breakdown, all this at the cheapest possible cost.

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 given positive integer. Thus, the network will be still functional when at most $k-1$ edges (nodes) fail.

However, this connectivity requirement may not unfortunately be sufficient to guarantee a high survivability and a routing quality. In fact, for some special networks such as VPN (virtual private networks), we may need a higher degree of connectivity. Moreover, the alternative routing path in the network may be too long and costly and this may cause a significant degradation in the transfer speed. In order to limit the rerouting length and guarantee a high QoS, it is commonly required that the length (number of edges) of the paths between an origin-destination pair is bounded by a given number L depending on technological parameters.

The problem is then to determine, given weights on the possible links of the network, and pairs of origin-destinations, a minimum weight network containing at least

✉ Ibrahima Diarrassouba
diarrasi@univ-lehavre.fr

Meriem Mahjoub
meriem.mahjoub@dauphine.fr

A. Ridha Mahjoub
ridha.mahjoub@lamsade.dauphine.fr

Hande Yaman
hyaman@bilkent.edu.tr

1 Normandie Univ, UNIHAVRE, LMAH, FR-CNRS-3335, 76600 Le Havre, France

2 PSL, CNRS UMR 7243 LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Cedex 16, Paris, France

3 Faculté des Sciences de Tunis, URAPOP, Université Tunis El Manar, UR13ZS38, Tunis, Tunisia

4 Department of Industrial Engineering, Bilkent University, Bilkent 06800 Ankara, Turkey

k edge (node) disjoint paths between each pair of origin-destination of length no more than L. This paper deals with the node connectivity case of the problem.

Consider an undirected graph $G = (V, E)$ with weights $c(e)$, $e \in E$, on the edges, an integer $L \geq 2$, and a set of demands $D \subset V \times V$. Each demand is an ordered pair (s, t) of nodes, with $s \neq t$. Node s is called the *source* (or *origin*) of the demand and t its *destination*. The *k-node-disjoint hop-constrained network design problem* (kNDHP for short) is to find a minimum weight subgraph of G containing at least k node-disjoint L-st-paths, that is, paths from s to t with at most L edges (also called hops), between each pair of nodes $(s, t) \in D$. The edge version of the problem has been widely studied in the literature. However, the kNDHP has been only considered for $k = 2$.

1.1 Node version with bounds

In [12], Diarrassouba et al. consider the kNDHP for $k = 2$. Here it is supposed that the two paths are node-disjoint and each path does not exceed L edges for a fixed integer $L \geq 1$. They investigate the structure of the associated polytope and describe several classes of valid inequalities when $L \leq 3$. Based on this, they devise a branch-and-cut algorithm. Huygens and Mahjoub [24] study the problem when $L = 4$ and $k = 2$. They show that the so-called cut and L-path-cut inequalities suffice for formulating the problem in this case. In [6], Chimani et al. consider $\{0, 1, 2\}$-survivable network design problems with node connectivity constraints. Given an edge-weighted graph and two customer sets $\mathscr{R}_1$ and $\mathscr{R}_2$, they look for a minimum cost subgraph that connects all customers, and guarantees 2-node connectivity for the $\mathscr{R}_2$ customers. They give a graph characterization of 2-node-connected graphs via orientation properties. Using this, they propose integer programming formulations based on directed graphs.

1.2 Edge version with bounds

The edge version of the problem has also been studied by several authors when $L = 2, 3$. In particular, in [26] Huygens et al. give a complete and minimal linear description of the corresponding polytope when $L = 2, 3$ and $|D| = 1$. In [25], Huygens et al. consider the problem when $|D| \geq 2$ and two edge-disjoint paths are required for each demand. They show that the problem is strongly NP-hard even when the demands in D are rooted at some node s and the costs are unitary. However, if the graph is complete, they prove that the problem in this case can be solved in polynomial time. They give an integer programming formulation of the problem in the space of the design variables when $L = 2, 3$, and they study the associated polytope. Moreover, they describe several classes of valid inequalities along with necessary and/or sufficient conditions to be facet defining, and propose a branch-and-cut algorithm.

In [2], Bendali et al. consider the more general k edge-disjoint hop-constrained problem (kEDHP) when k edge-disjoint paths are required. They discuss a branch-and-cut algorithm for the problem when $L = 2, 3$. Huygens and Mahjoub [24] study the kEDHP when $L = 4$ and $k = 2$. They introduce a new general class of valid inequalities. Using this, they give an integer programming formulation of the problem in the natural space of variables. In [7], Dahl considers the hop-constrained path problem, that is the problem of finding between two distinguished nodes s and t a minimum cost path with no more than L edges when L is fixed. He gives a complete description of the dominant of the associated polytope when $L \leq 3$ and a class of facet defining inequalities for $k \geq 4$. Dahl and Gouveia [9] consider the directed hop-constrained shortest path problem. They describe valid inequalities and characterize the associated polytope when $L = 2, 3$. A related problem is considered in Dahl et al. [8], the hop-constrained walk problem. The authors discuss the associated polytope in directed graphs when $L = 4$.

In [18], Gouveia and Leitner consider the network design problem with vulnerability constraints. The solutions to the problem are subgraphs containing a path of length at most H_{st} for each commodity $\{s, t\}$ and a path of length at most H'_{st} between s and t after at most $k - 1$ edge failures. They give characterizations of feasible solutions and propose integer programming formulations. In [19], Gouveia et al. consider the problem with bounded lengths in the context of an MPLS (multi-protocol label switching) network design model. They discuss two models involving one set of variables associated to each path between each pair of demand nodes (a standard network flow model with additional cardinality constraints and a model with hop-indexed variables) and a third model involving one single set of hop-indexed variables for each demand pair. They show that the aggregated more compact hop-indexed model produces the same linear programming bound as the multi-path hop-indexed model.

1.3 Extended formulations for the edge version with bounds

In [4], Botton et al. consider the hop-constrained survivable network design problem with reliable edges, i.e., edges that are not subject to failure. They study two variants, a static problem where the reliability of the edges is given and an upgrading problem where edges can be upgraded to the reliable status at a given cost. They adapt for the two variants an extended formulation proposed in Botton et al. [5] for the case without reliable edges. They use

Benders decomposition to accelerate the solution process. Their computational results indicate that these two variants appear to be more difficult to solve than the original problem (without reliable edges). In [32], Mahjoub et al. propose an extended formulation for the rooted case, when all the demands have a common vertex, called hop-level multicommodity flow formulation, inspired from the formulation given in [5]. The authors introduce the concept of solution level. To each solution of the problem, a partition of the node set into $L+2$ levels can be associated according to the distance to the root in the solution. Then, they reduce the problem to a specific multicommodity flow problem in an auxiliary layered directed graph.

In Table 1, a summary of the previously studied hop-constrained network design problems is presented.

1.4 Edge and node versions without bounds

The k-node-connected subgraph problem without bounds on the paths has been considered in the literature. In [31], Mahjoub and Nocq discuss structural properties of the 2-node-connected polytope (see also [1]). Grötschel et al. [20–23] study the problem within a more general survivability model. In [21], 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 view and propose cutting plane algorithms [21–23]. In [28], Kerivin et al. propose branch-and-cut algorithms for both versions of the $\{1,2\}$ survivable network design problem. Here, the type of each node is either 1 or 2. In [29], Mahjoub et al. consider the k-node-connected subgraph problem. They give valid inequalities and propose a branch-and-cut algorithm.

The uniform edge case without hop constraints has been widely investigated. The reader can be referred to [3, 14, 15, 20–23] for more details.

In Table 2, we show the studied survivable network models with node versus edge connectivity.

As indicated in Table 2, the kNDHP has not been considered for $k \geq 3$. The aim of this paper is to discuss this case for $L = 2, 3$ from a polyhedral point of view. We present new valid inequalities along with separation algorithms. We discuss conditions for these inequalities to define facets. Using these results, we propose a branch-and-cut algorithm for solving the problem in this case.

In the rest of this section, we give some notations. We will denote an undirected graph by $G = (V, E)$ where V is the *node set* and E is the *edge set*. Given a set of nodes $Z \subset V$, we denote by $G \setminus Z$ the subgraph obtained from G by deleting the nodes in Z and all their incident edges. For $W \subseteq V$, we let $\overline{W} = V \setminus W$. The set $\delta_G(W)$ 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 disjoint node subsets $W_1, \ldots, W_p \subset V$, $p \geq 2$, we denote by $\delta_G(W_1, \ldots, W_p)$ the set of edges of G between the sets $W_1, \ldots, W_p$. And we will denote by $[V_i, V_j]$ the set of edges between V_i and V_j. Given $F \subseteq E$, $c(F)$ will denote $\sum_{e \in F} c(e)$ and the *incidence vector* of F, denoted by x^F, is the $0-1$ vector which takes 1 if $e \in F$ and 0, if not.

Table 1 State of the art of the hop-constrained survivable network design problem

Connectivity	Type of paths	Reference	Results
$k = 1$	–	Dahl and Gouveia [9]	Valid inequalities for the directed hop-constrained shortest path problem. Complete linear characterizations of the hop-constrained path polytope when $L = 2, 3$
$k = 2$	Edge/node-disjoint	Huygens and Mahjoub [24]	IPF in the space of the design variables, for the node case when $L \leq 4$
$k = 2$	Edge/node-disjoint	Huygens et al. [25]	IPF, valid inequalities and branch-and-cut algorithm for $L = 2, 3$
$k \geq 1$	Edge-disjoint	Bendali et al. [2]	Characterization of the associated polytope for $L = 3$ and $\|D\| = 1$
$k \geq 1$	Edge-disjoint	Diarrassouba et al. [13]	Valid inequalities and branch-and-cut and branch-and-cut-and-price algorithms for $L = 2, 3$
$k = 2$	Node-disjoint	Diarrassouba et al. [12]	Valid inequalities and branch-and-cut algorithm for $L = 3$

Table 2 Models of survivable networks with node versus edge connectivity

Connectivity	Bound	Edge case results	Node case results
$k = 2$	$L = \infty$	ILP formulation, valid inequalities, separation, branch-and-cut, polytope characterization [1, 20, 22, 23, 27, 30]	ILP formulation, valid inequalities, separation, branch-and-cut [20, 22, 23, 27, 31]
$k \geq 3$	$L = \infty$	ILP formulation, valid inequalities, separation, branch-and-cut, polytope characterization [1, 20, 23, 27, 30]	ILP formulation, separation valid inequalities, branch-and-cut [3, 8, 20, 23]
$k = 2$	$L = 2, 3$	ILP formulation, valid inequalities, separation, branch-and-cut [25, 26]	ILP formulation, valid inequalities polyhedral study, branch-and-cut [2, 8, 20, 23]
$k = 2$	$L = 4$	ILP formulation, valid inequalities, separation, branch-and-cut [24, 25]	ILP formulation, valid inequalities branch-and-cut [24]
$k \geq 3$	$L = 2, 3$	ILP formulation, valid inequalities, separation, branch-and-cut, extended formulation [2, 4, 5, 7–9, 13]	Considered in this paper

The remaining of the paper is organized as follows. In Section 2, we give an integer programming formulation for the problem. In Section 3, we investigate the kNDHP polytope and present several classes of valid inequalities. Then, in Section 4, we discuss conditions under which these inequalities define facets. Using these results, we propose, in Sections 5 and 6, branch-and-cut algorithms for the problem when $k \geq 3$ and $L = 3$, and when $L = 4$ and $k = 2$, respectively, and present computational results. Finally, we give some concluding remarks in Section 7.

2 Integer programming formulation

Let $G = (V, E)$ be a graph and $F \subseteq E$ an edge set which induces a solution of the kNDHP. As F is a solution of the problem, the subgraph induced by F, say G_F, contains k edge-disjoint st-paths for every $(s, t) \in D$. Thus, by Menger's theorem [33], every st-cut of G_F contains at least k edges. Consequently, the incidence vector of F satisfies the following inequalities

$$x(\delta_G(W)) \geq k, \qquad \text{for all st-cut } \delta(W) \text{ and } (s, t) \in D. \quad (1)$$

Inequalities (1) are called *st-cut inequalities*.

Dahl [7] introduces a class of valid inequalities as follows.

Let $(V_0, \ldots, V_{L+1})$ be a partition of V with $s \in V_0$, $t \in V_{L+1}$, and $V_i \neq \emptyset$ for all $i \in \{1, \ldots L\}$. Let T be the set of edges $uv \in E$ such that $u \in V_i$, $v \in V_j$ and $|i - j| > 1$, that is,

$$T = \delta(V_0, \ldots, V_{L+1}) \setminus \bigcup_{i=0}^{L} [V_i, V_{i+1}].$$

The set T is called an *L-st path-cut*. Then, the inequality

$$x(T) \geq 1$$

is valid for the L-st-path polyhedron. Using similar type of partitions, we can generalize these inequalities to the kNDHP as

$$x(T) \geq k, \quad \text{for every } L\text{-st-path-cut } T \text{ of } G, \text{ for any } (s, t) \in D. \quad (2)$$

Inequalities of type (2) are called *L-st-path-cut inequalities* (Fig. 1).

Inequalities (1) and (2) can be easily adapted in order to ensure the existance of k node-disjoint paths of length at most L. Given node subsets $Z \subset V \setminus \{s, t\}$ for $(s, t) \in D$, and $W \subset V \setminus Z$, the *st node-cut* $\delta_{G\setminus Z}(W)$ of G is the st-cut induced by W in $G \setminus Z$. Any L-st path-cut in $G \setminus Z$ is called an *L-st-node path-cut* of G.

A solution $x \in \mathbb{R}^E$ of the kNDHP also satisfies the following inequalities

$$x(\delta_{G\setminus Z}(W)) \geq k - |Z|, \quad \text{for all st-node-cut } \delta_{G\setminus Z}(W), \ Z \subset V \setminus \{s, t\} \text{ such that } 1 \leq |Z| \leq k - 1, \text{ and } (s, t) \in D, \quad (3)$$

$$x(T_{G\setminus Z}) \geq k - |Z|, \quad \text{for all } L\text{-st-node-path-cut } T_{G\setminus Z} \text{ of } G \setminus Z, Z \subset V \setminus \{s, t\} \text{ such that } 1 \leq |Z| \leq k - 1, \text{ and } (s, t) \in D. \quad (4)$$

Inequalities (3) and (4) are called, respectively, *st-node-cut* and *L-st-node-path-cut inequalities*. Moreover, the

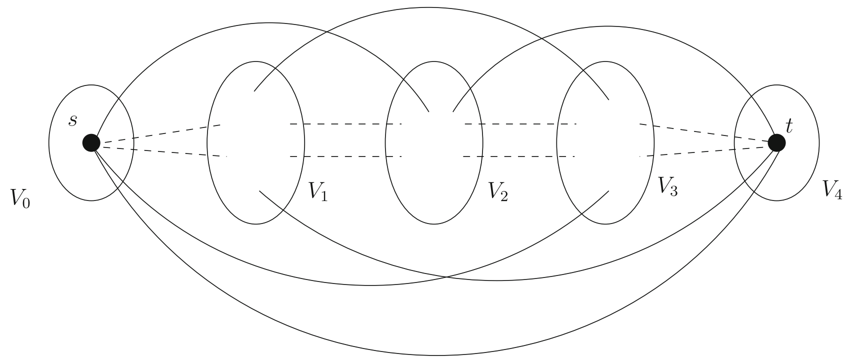

Fig. 1 Support graph of an L-st-path-cut with $L = 3$ and T formed by the solid edges

incidence vector of an edge set F inducing a solution of the kNDHP satisfies

$$x(e) \geq 0, \text{ for all } e \in E, \quad (5)$$

$$x(e) \leq 1, \text{ for all } e \in E. \quad (6)$$

In the following, we show that the st-cut, st-node-cut, L-st-path-cut, L-st-node-path-cut, and trivial inequalities, together with integrality constraints, suffice to formulate the kNDHP as a $0-1$ linear program when $L \in \{2, 3\}$.

For this, we consider, for each demand (s, t) the directed $\widetilde{G}_{st} = (\widetilde{V}_{st}, \widetilde{A}_{st})$ obtained as follows (see also [2] and [11]). The node set $\widetilde{V}_{st}$ is formed by the nodes s, t, the node set of $V \setminus \{s, t\}$ and a copy u' for each node $u \in V \setminus \{s, t\}$. The set of arcs $\widetilde{A}_{st}$ is obtained as follows. For each edge su $\in E$ (resp. ut $\in E$), we add in $\widetilde{A}_{st}$ an arc (s, u') (resp. (u', t)). For each edge uv $\in E$, with $u, v \neq s, t$, we add two arcs (u, v') and (v, u') in $\widetilde{A}_{st}$. Finally, for each node $u \in V \setminus \{s, t\}$, we add an arc (u, u') in $\widetilde{A}_{st}$. It is not hard to see that every st-dipath of $\widetilde{G}_{st}$ corresponds to a 3-st-path of G, and vice-versa. Also, two node-disjoint 3-st-path of G correspond to two directed node-disjoint st-path of $\widetilde{G}_{st}$. However, the converse is not true, that is two node-disjoint st-dipaths of $\widetilde{G}_{st}$ may not correspond to node-disjoint 3-st-paths of G (see Fig. 2 for illustration).

Bendali et al. [2] show that every st-cut and 3-st-path-cut $C \subseteq E$ can be associated with a directed st-cut $\widetilde{C} \subseteq \widetilde{A}_{st}$ which does not contain an arc of the form (u, u'), with $u \in V \setminus \{s, t\}$, and vice-versa. Moreover, they show that a solution $\overline{x} \in \mathbb{R}^E$ can be associated with a solution $\overline{y} \in \mathbb{R}^{\widetilde{A}_{st}}$ such that $\overline{x}(C) = \overline{y}(\widetilde{C})$.

Now, we give the following theorem.

Theorem 1 *Let $\overline{x} \in \{0, 1\}^E$ be an integral solution, which satisfies all the cut and 3-st-path-cut inequalities (1) and (2). Then, $\overline{x}$ induces a solution of the kNHDP if and only if it satisfies all the st-node-cut and 3-st-node-path-cut inequalities.*

Proof As the st-node-cut and the 3-st-node-cut inequalities are valid for the kNDHP, if $\overline{x}$ is a solution of the kNDHP, then it satisfies these inequalities.

Now suppose that $\overline{x}$ does not induce a feasible solution of the kNHDP, that is the subgraph of G induced by $\overline{x}$, denoted by $G(\overline{x}) = (V, E(\overline{x}))$ does not contain k node-disjoint 3-st-paths for some demand $(s, t) \in D$. We are going to show that there exists an st-node-cut or a 3-st-node-path-cut inequality which is violated by $\overline{x}$.

Let $\widetilde{G}_{st}$ be the directed graph associated with (s, t) as described above, and let $\widetilde{y} \in \mathbb{R}^{\widetilde{A}_{st}}$ be a weight vector such that

$$\widetilde{y}(a) = \begin{cases} 1 & \text{if } a \text{ corresponds to edge } e \text{ and } e \in E(\overline{x}), \\ 0 & \text{if } a \text{ corresponds to edge } e \text{ and } e \notin E(\overline{x}), \\ +\infty & \text{if } a = (u, u') \text{ for all } u \in V \setminus \{s, t\}. \end{cases}$$

Remark that, as G is simple, that it is does not contain parallel edges, if two 3-st-paths P_1 and P_2 are not node-disjoint, then they are of the form $P_1 = (s, u, v, t)$ and $P_2 = (s, v, z, t)$ with $u, v, z \in V \setminus \{s, t\}$ and $u \neq v \neq z$. These two paths correspond in $\widetilde{G}_{st}$ to paths (s, u, v', t) and (s, v, z', t). Conversely, two paths (s, u, v', t) and (s, v, z', t) of $\widetilde{G}_{st}$ correspond to two paths (s, u, v, t) and (s, v, z, t) which are not node-disjoint. Consequently, when $\overline{x}$ is not feasible for

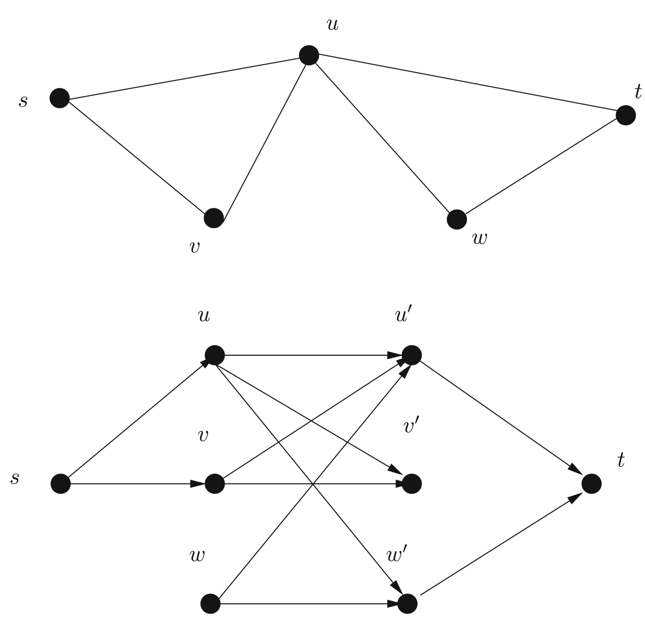

Fig. 2 Construction of the graph H for $L = 3$

the kNHDP, any maximum set of disjoint st-dipaths of the graph $\widetilde{G}_{st}$, will contain two paths of the form (s, u, v', t) and (s, v, z', t), with $u, v, z \in V \setminus \{s, t\}$ and $u \neq v \neq z \neq u$.

Now we introduce the following procedure, that we call *Procedure BuildZ*, which aims to build a node set $Z \subseteq V$, from which we will obtain the violated st-node-cut or 3-st-node-path-cut inequalities. Let $Z \subseteq V$ be a node set of G and denote by $\widetilde{Z}$ the nodes of $\widetilde{G}_{st}$ corresponding to those of Z, that is $\widetilde{Z} = \{u, u'$ such that $u \in Z\}$. At the begining of the procedure $Z = \emptyset$. Now compute a maximum flow from s to t in $\widetilde{G}_{st} \setminus \widetilde{Z}$, with each arc $a \in \widetilde{A}_{st}$ having the capacity $\widetilde{y}(a)$. This gives a maximum set $\widetilde{\mathcal{P}}$ of node-disjoint st-dipaths in $\widetilde{G}_{st} \setminus \widetilde{Z}$, as the flow going in or out of a node $v \in W \setminus \{s', t'\}$ is either 0 or 1. Indeed, each node $v \in W \setminus \{s', t'\}$ has at most one arc going in from s' and at most one arc going out to t'. Remark that some of these paths may correspond to non node-disjoint 3-st-paths of G, that is they are of the form (s, u, v', t) and (s, v, z', t). Let $\widetilde{\mathcal{P}}'$ be the set of these paths. Also let $\mathcal{P}'$ be the set of paths of G corresponding to those of $\widetilde{\mathcal{P}}'$ and $U \subseteq V$ the set of nodes of G which are shared by two paths of $\mathcal{P}'$. Now, add to Z the nodes of U and repeat this procedure until $|Z| \geq k$ or $U = \emptyset$. It should be noticed that when $U = \emptyset$, the arc-disjoint st-dipaths obtained by the computation of the maximum flow in $\widetilde{G}_{st} \setminus \widetilde{Z}$ correspond to node-disjoint 3-st-paths of $G \setminus Z$.

The identification of the nodes of U can be easily done by simply considering, for each node $u \in V \setminus Z$, the arcs entering and leaving nodes u and u' with flow value 1. Namely, consider a node $v \in U$. This means that after the maximum flow computation, there are two paths (s, u, v', t) and (s, v, z', t). Since the arc capacities are either 0 or 1, this means that

- the flow value on arc (s, v) is 1,
- the flow value on arc (v, v') is 0,
- the flow value on arc (v', t) is 1.

Figure 3 illustrates the above remark. The solid lines represent arcs having flow value 1 and dashed lines represent arcs with flow value 0. The flow value of the arcs represented by dotted lines may be 0 or 1.

Thus, let $Z \subseteq V \setminus \{s, t\}$ be the node set obtained by the application of procedure BuildZ. It is not hard to see that by the construction of Z, the graph $G(\overline{x})$ contains $|Z|$ st-paths of the form (s, u, t), for all $u \in Z$. Clearly, these paths are node-disjoint. This also implies that $|Z| \leq k - 1$, for otherwise, $G(\overline{x})$ would contain at least k node-disjoint 3-st-paths, which is not possible. Now compute a maximum flow from s to t in $\widetilde{G}_{st} \setminus \widetilde{Z}$, and let f be the value of that flow. By the construction of Z, this later flow corresponds to a set of f disjoint 3-st-paths of G which are node-disjoint. Moreover, these paths are node-disjoint from those induced by Z. Thus, together with the paths induced by Z, we obtain $|Z| + f$ node-disjoint 3-st-paths in $G(\overline{x})$. As by assumption,

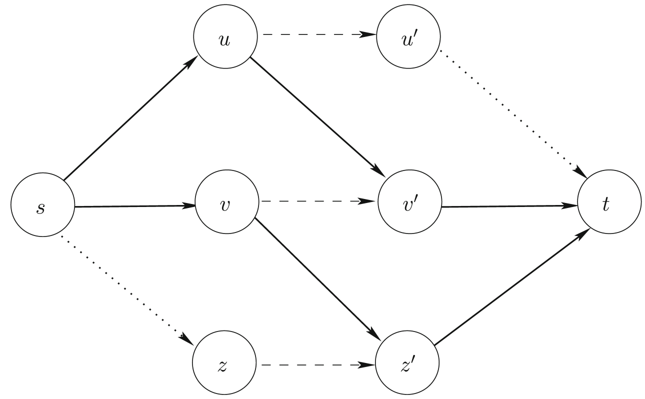

Fig. 3 Two st-dipaths of $\widetilde{G}_{st} \setminus \widetilde{Z}$ inducing non node-disjoint 3-st-paths in $G \setminus Z$

$G(\overline{x})$ does not contain k node-disjoint 3-st-paths, we have that $|Z| + f < k$, that is $f < k - |Z|$.

Now, as f is the value of the maximum flow of $\widetilde{G}_{st} \setminus Z$, the weight of a minimum cut $\widetilde{C}$ of $\widetilde{G}_{st} \setminus Z$ is $\widetilde{y}(\widetilde{C}) = f < k - |Z|$. Finally, as shown by Bendali et al. [2], $\widetilde{C}$ corresponds to an edge set C which is either an st-cut or a 3-st-path-cut of $G \setminus Z$, that is C corresponds to an st-node-cut or a 3-st-node-path-cut of G whose weight is $\overline{x}(C) = \widetilde{y}(\widetilde{C}) = f < k - |Z|$. Consequently, the st-node-cut or 3-st-node-cut induced by C is violated by $\overline{x}$. □

From Theorem 1 the kNDHP is equivalent to

$$\min\{cx | x \text{ satisfies } (1) - (6) \text{ and } x \in \mathbb{Z}_+^E\}. \tag{7}$$

We will call inequalities (1)–(6) *basic inequalities*. Here, *basic* means that they are necessary in the basic formulation of the problem. We will denote by kNDHP(G, L) the convex hull of all the integer solutions of Eqs. 1–6, and call kNDHP(G, L) *the k-node-disjoint hop-constrained problem polytope*.

Formulation (7) is no longer valid for $L \geq 5$. Consider for example the graph shown in Fig. 4. For $k = 2$, its incidence vector satisfies inequalities (1)–(6) but the graph does not contain two node-disjoint st-paths of length at most $L = 5$. This example is borrowed from [24].

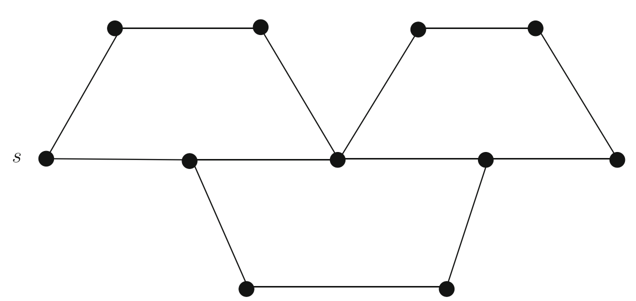

Fig. 4 Infeasible solution of the 2NDHP with $L = 5$ and $D = \{(s, t)\}$

3 Polytope and valid inequalities

In this section, we present several classes of valid inequalities inspired from the kEDHP (k-edge-disjoint hop-constrained problem) that have been introduced in the literature. Since any solution of the kNDHP is also solution of the kEDHP, any valid inequality for the kEDHP polytope on G is also valid for kNDHP(G, L). Also note that if $S \subseteq E$ is a solution of kNDHP in G and $Z \subset V$, such that $|Z| \leq k-1$, then the restriction of S on $G \setminus Z$ is a solution of the $(k-|Z|)$NDHP on $G \setminus Z$ with respect to origin-destination pairs contained in $G \setminus Z$.

Lemma 1 *Let $Z \subset V$, and let $D' \subseteq D$ be a subset of origin-destination pairs in $G \setminus Z$. Suppose that $y \neq \emptyset$. If an inequality $ax \geq \alpha(k)$ is valid for kNDHP(G, L) in G with respect to D then the inequality $yx \geq \alpha(k-|Z|)$ is valid for kNDHP$(G \setminus Z, L)$, with respect to D', where a' is the restriction of a on $G \setminus Z$.*

Note that in Lemma 1, we consider $\alpha(k)$ as a right-hand side in the inequality $ax \geq \alpha(k)$ just to express the fact that the right-hand side of a valid inequality of the kNDHP may depend of k.

3.1 Generalized L-st-path-cut inequalities

Dahl and Gouveia [9] introduce the so-called generalized L-st-path-cut inequalities for the problem of finding an L-st-path between two nodes s and t. They are defined as follows. Let $(s,t) \in D$ and $\pi = (V_0, \ldots, V_{L+r})$, $r \geq 1$, be a partition of V such that $s \in V_0$ and $t \in V_{L+r}$. Then, the generalized *L-st-path-cut inequality* induced by (s,t) and π is

$$\sum_{e \in [V_i, V_j], i \neq j} \min(|i-j|-1, r) x(e) \geq r. \tag{8}$$

These inequalities can be easily extended to the kNDHP by replacing the right-hand-side of inequality (8) by $(k-|Z|)r$, with $Z \subset V$, $|Z| \leq k-1$, yielding

$$\sum_{e \in [V_i, V_j], i \neq j} \min(|i-j|-1, r) x(e) \geq (k-|Z|)r. \tag{9}$$

Inequality (9) is valid for kNDHP(G, L). A *jump* is an edge between two non-consecutive sets of π. Inequality (9) gives the minimum number of jumps in a partition $\pi = (V_0, \ldots, V_{L+r})$ needed in a solution of the problem. Inequalities of type (9) will also be called *generalized L-st-path-cut inequalities*.

3.2 Double cut inequalities

Huygens et al. [25] introduce the so-called *double cut inequalities* for the 2EDHP for $L = 3$. They are defined as follows. Consider the partition $\pi = (V_0^1, V_0^2, V_1, \ldots, V_4)$ of V such that $(V_0^1, V_0^2 \cup V_1, V_2, V_3, V_4)$ induces a 3-st-path-cut, and V_1 induces a valid st-cut in G. If $F \subseteq [V_0^2 \cup V_1 \cup V_4, V_2]$ is chosen such that $|F|$ is odd, then the double cut inequality can be written as follows:

$$x([V_0^1, V_1 \cup V_2 \cup V_3 \cup V_4]) + x([V_0^2, V_1 \cup V_3 \cup V_4]) \\ +x([V_1, V_3 \cup V_4]) + x([V_0^2 \cup V_1 \cup V_4, V_2]) \geq \left\lceil 3 - \frac{|F|}{2} \right\rceil \tag{10}$$

We now generalize these inequalities for the kNDHP for $L \geq 2$. Let $Z \subset V \setminus \{s,t\}$, for $(s,t) \in D$, and $V_0, \ldots, V_{i_0-1}, V_{i_0}^1, V_{i_0}^2, V_{i_0+1}, \ldots, V_{L+1}$ be a family of node subsets of $V \setminus Z$ such that $\pi = (V_0, \ldots, V_{i_0-1}, V_{i_0}^1, V_{i_0}^2 \cup V_{i_0+1}, \ldots, V_{L+1})$ induces a partition of $G \setminus Z$ (see Fig. 5 for illustration). Suppose that

1. there exists an $(s,t) \in D$ such that $V_{i_0}^1 \cup V_{i_0}^2$ induces an st-node-cut in $G \setminus Z$ and $s \in V_{i_0}^1$ or $t \in V_{i_0}^1$,
2. there exists an $(s,t) \in D$ such that V_{i_0+1} induces an st-node-cut in $G \setminus Z$,
3. there exists an $(s,t) \in D$ such that π induces an L-st-node-path-cut in $G \setminus Z$ with $s \in V_0$ (resp. $t \in V_0$) and $t \in V_{L+1}$ (resp. $s \in V_{L+1}$).

Let $\overline{E} = [V_{i_0-1}, V_{i_0}^1] \cup [V_{i_0+2}, V_{i_0}^2 \cup V_{i_0+1}] \cup \left(\bigcup_{k,l \notin \{i_0, i_0+1\}, |k-l|>1} [V_k, V_l]\right)$ and $F \subseteq \overline{E}$ such that $|F|$ and $k-|Z|$ have different parities.

Let also $\hat{E} = \left(\bigcup_{i=0}^{i_0-2} [V_i, V_{i+1}]\right) \cup \left(\bigcup_{i=i_0+2}^{L} [V_i, V_{i+1}]\right) \cup F$.

Then, we have the following inequality.

$$x(\delta(\pi) \setminus \hat{E}) \geq \left\lceil \frac{3(k-|Z|)-|F|}{2} \right\rceil \tag{11}$$

Theorem 2 *Inequalities (11) are valid for kNDHP(G, L).*

Proof Let $T_{G \setminus Z}$ be the L-st-node-path-cut of $G \setminus Z$ induced by the partition π and Z. Thus, the following inequalities are valid for kNDHP(G, L),

$$\begin{aligned} x_{G \setminus Z}(T) &\geq k-|Z|, \\ x(\delta_{G \setminus Z}(V_{i_0}^1 \cup V_{i_0}^2)) &\geq k-|Z|, \\ x(\delta_{G \setminus Z}(V_{i_0+1})) &\geq k-|Z|, \\ -x(e) &\geq -1 \text{ for all } e \in F, \\ x(e) &\geq 0 \quad \text{for all } e \in \overline{E} \setminus F. \end{aligned} \tag{12}$$

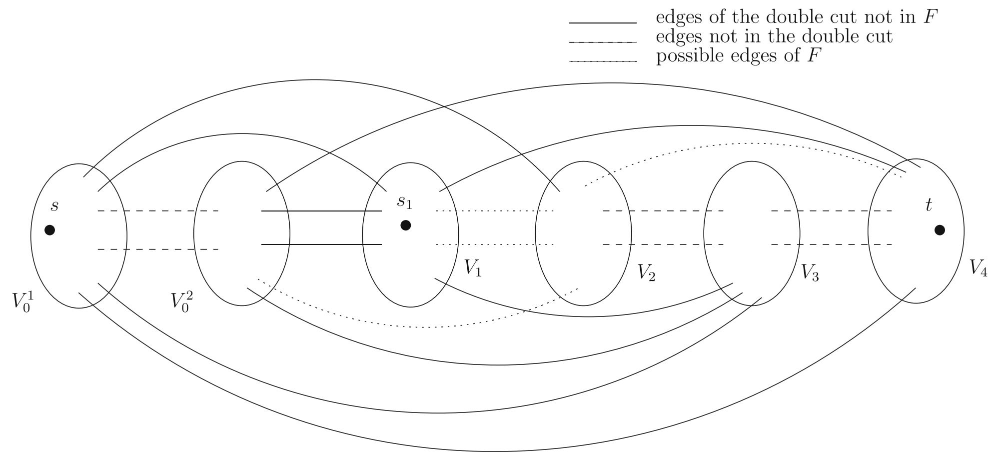

Fig. 5 A double cut with $L = 3$ and $t_1 = t$

By summing these inequalities, dividing by 2 and rounding up the right-hand side, we obtain inequality (11). □

These inequalities will also be called *double cut inequalities*.

If $L = 3$ and $i_0 = 0$, inequality (11) can be written as follows:

$$x([V_0^1, V_1 \cup V_2 \cup V_3 \cup V_4]) + x([V_0^2, V_1 \cup V_3 \cup V_4]) \\ +x([V_1, V_3 \cup V_4]) + x([V_0^2 \cup V_1 \cup V_4, V_2] \setminus F) \\ \geq \left\lceil \frac{3(k - |Z|) - |F|}{2} \right\rceil. \quad (13)$$

Here, $\pi = (V_0^1, V_0^2 \cup V_1, V_2, V_3, V_4)$ and $F \subseteq [V_0^2 \cup V_1 \cup V_4, V_2]$ such that $|F|$ and $k - |Z|$ have different parities.

3.3 Triple path-cut inequalities

Huygens et al. [25] introduce the so-called *triple path-cut inequalities* for the 2EDHP for $L = 3$. They are defined for a partition $(V_0, V_1, \ldots, V_5)$ of V with $s_1, s_2 \in V_0$, $t_1 \in V_4$ and $t_2 \in V_5$. Then, the triple path-cut inequality

$$2x([V_0, V_2]) + 2x([V_0, V_3]) + 2x([V_1, V_3]) \\ +x([V_0 \cup V_1 \cup V_2 \cup V_3, V_4 \cup V_5] \setminus \{e\}) + x([V_4, V_5]) \geq 3 \quad (14)$$

where $e \in [V_2 \cup V_3, V_4] \cup [V_3, V_5]$, is valid for 2EDHP$(G, 3)$.

We now generalize these inequalities for the kNDHP for $L = 3$.

Theorem 3 *Let $Z \subset V \setminus R_D$, where R_D is the set of terminal nodes of G. Let $(V_0, \ldots, V_3, V_4^1, V_4^2, V_5^1, V_5^2)$ be a family of node sets of $V \setminus Z$ such that $(V_0, \ldots, V_3, V_4^1 \cup V_4^2, V_5^1 \cup V_5^2)$ induces a partition of $V \setminus Z$ and there exist two demands $\{s_1, t_1\}$ and $\{s_2, t_2\}$ with $s_1, s_2 \in V_0$, $t_1 \in V_4^2$ and $t_2 \in V_5^2$. The sets V_4^1 and V_5^1 may be empty and s_1 and s_2 may be the same. Let also $V_4 = V_4^1 \cup V_4^2$, $V_5 = V_5^1 \cup V_5^2$ and $F \subseteq [V_2, V_4^2] \cup [V_3, V_4 \cup V_5]$ such that $|F|$ and $k - |Z|$ have different parities. Then, the inequality*

$$2x([V_0, V_2]) + 2x([V_0, V_3]) + 2x([V_1, V_3]) \\ +x([V_0 \cup V_1, V_4 \cup V_5]) + x([V_4, V_5]) \\ +x([V_2, V_5]) + x(([V_2, V_4] \cup [V_3, V_4 \cup V_5]) \setminus F) \\ \geq \left\lceil \frac{3(k - |Z|) - |F|}{2} \right\rceil \quad (15)$$

is valid for kNDHP$(G, 3)$.

Proof Let T_1 be the 3-s_1t_1-node-path-cut induced by the partition $(V_0, V_1 \cup V_5, V_2, V_3 \cup V_4^1, V_4^2)$ and Z, and T_2 and T_3 be the 3-s_2t_2-node-path-cuts induced by the partitions $(V_0, V_1 \cup V_4, V_2, V_3 \cup V_5^1, V_5^2)$ and $(V_0, V_1, V_2, V_3 \cup V_4 \cup V_5^1, V_5^2)$, respectively, and Z. The following inequalities are valid for kNDHP$(G, 3)$.

$$\begin{aligned} x_{G\setminus Z}(T_1) &\geq k - |Z|, \\ x_{G\setminus Z}(T_2) &\geq k - |Z|, \\ x_{G\setminus Z}(T_3) &\geq k - |Z|, \\ -x(e) &\geq -1 \quad \text{for all } e \in F, \\ x(e) &\geq 0 \text{ for all } e \in ([V_2, V_4^2] \cup [V_3, V_4 \cup V_5]) \setminus F. \end{aligned} \quad (16)$$

By summing these inequalities, dividing by 2 and rounding up the right-hand side, we obtain inequality (15). □

Inequalities (15) will also be called *triple-path-cut inequalities*. Figure 6 gives an illustration.

3.4 Steiner partition inequalities

Let R_D be the set of terminal nodes of G. Let $Z \subset V \setminus R_D$, and $(V_0, V_1, \ldots, V_p)$, $p \geq 2$, be a partition of $V \setminus Z$

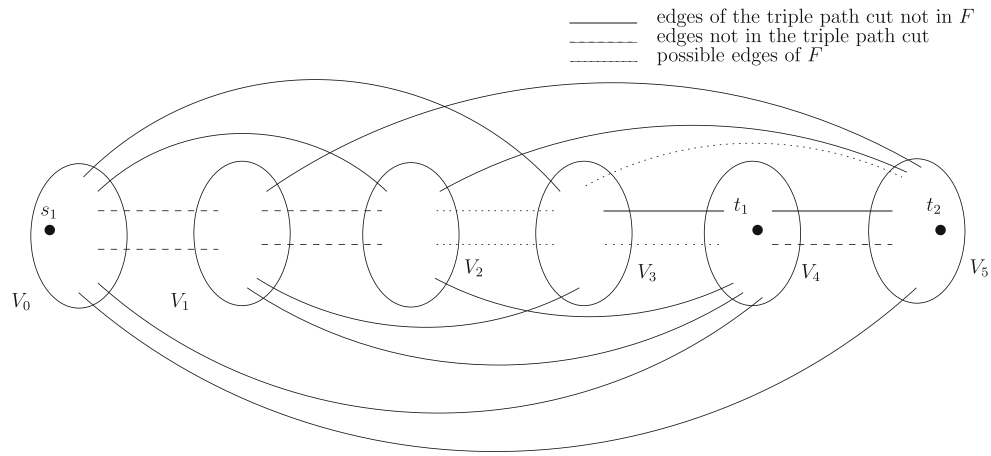

Fig. 6 A triple path-cut with $k = 2$, $L = 3$ and $s_1 = s_2$

such that $V_0 \subseteq V \setminus R_D$, and for all $i \in \{1, \ldots, p\}$ there is a demand $\{s, t\} \in D$ such that V_i induces an st-cut of G. We can see that V_0 may be empty. The partition $(V_0, V_1, \ldots, V_p)$ is called a *Steiner partition*. And we have the following inequality

$$x(\delta_{G\setminus Z}(V_0, \ldots, V_p)) \geq \left\lceil \frac{(k - |Z|)p}{2} \right\rceil. \quad (17)$$

Inequalities of type (17) will be called *Steiner partition inequalities*.

Theorem 4 *Inequalities (17) are valid for kNDHP(G, L).*

Proof The following inequalities are valid for kNDHP(G, L).

$$x_{G\setminus Z}(V_i) \geq k - |Z|, \text{ for } i = 1, \ldots, p,$$
$$x(e) \geq 0, \quad \text{for all } e \in \delta(V_0). \quad (18)$$

By adding them, we obtain

$$2x(\delta_{G\setminus Z}(V_0, \ldots, V_p)) \geq (k - |Z|)p.$$

By dividing by 2 and rounding up the right-hand side, we get inequality (17). □

3.5 Steiner SP-partition inequalities

Diarrassouba et al. [13] introduced the so-called Steiner SP-partition inequalities for the kEDHP. In what follows, we extend these inequalities to the kNDHP. They are defined as follows. Let $Z \subset V \setminus R_D$, where R_D is the set of terminal nodes of G. Consider a partition $\pi = (V_1, \ldots, V_p)$, $p \geq 3$, of $V \setminus Z$, such that the graph $G_\pi = (V_\pi, E_\pi)$ is series-parallel (G_π is the graph obtained by contracting the sets V_i, $i = 1, \ldots p$). Suppose that $V_\pi = \{v_1, \ldots, v_p\}$ where v_i is the node of G_π obtained after the contraction of the set V_i, $i = 1, \ldots, p$. The partition π is called a *Steiner SP partition* if and only if π is a Steiner partition and either

1. $p = 3$ or
2. $p \geq 4$ and there exists a node $v_{i_0} \in V_\pi$ incident to exactly two nodes v_{i_0-1} and v_{i_0+1} such that after the contraction of the sets V_{i_0}, V_{i_0-1} and V_{i_0}, V_{i_0+1}, the partitions π_1 and π_2 obtained from π are also Steiner-SP-partitions.

Theorem 5 [13] *Let $\pi = (V_1, \ldots, V_p)$, $p \geq 3$, be a partition of V such that G_π is series-parallel. The partition π is a Steiner-SP-partition of G if and only if the subgraph of G_D induced by π is connected.*

From Theorem 5, note that if the demand graph is connected, then every Steiner partition of $V \setminus Z$ inducing a series-parallel subgraph of $G \setminus Z$ is a Steiner-SP -partition of $V \setminus Z$. With a Steiner-SP-partition $(V_1, \ldots, V_p)$, $p \geq 3$, we associate the following inequality

$$x(\delta_{G\setminus Z}(V_1, \ldots, V_p)) \geq \left\lceil \frac{k - |Z|}{2} \right\rceil p - 1. \quad (19)$$

Inequalities of type (19) are called *Steiner-SP-partition inequalities*.

Theorem 6 *Inequalities (19) are valid for kNDHP(G, L).*

Proof Let $\pi = (V_1, \ldots, V_p)$ be a Steiner-SP-partition. The proof is by induction on p. If $p = 3$, as π is a Steiner partition then we associate with π the inequality

$$x(\delta_{G\setminus Z}(V_1, V_2, V_3)) \geq \left\lceil \frac{3(k - |Z|)}{2} \right\rceil = 3 \left\lceil \frac{(k - |Z|)}{2} \right\rceil - 1. \quad (20)$$

Now suppose that every inequality (19) induced by a Steiner-SP-partition of p elements, $p \geq 3$, is valid for kNDHP$(G, 3)$ and let $\pi = (V_1, \ldots, V_p, V_{p+1})$ be a Steiner-SP-partition. Since G_π is series-parallel, then there exists a node set V_{i_0} of π such that it is incident to exactly two elements of π, V_{i_0-1} and V_{i_0+1}. Let $T_1 = [V_{i_0}, V_{i_0+1}]$ and

$T_2 = [V_{i_0}, V_{i_0-1}]$. As π is a Steiner-SP-partition, it is also a Steiner partition. As V_{i_0} and $Z \subset V \setminus \{s,t\}$ induce a valid st-node-cut inequality, for some $\{s,t\} \in D$. Thus, $x(T_1) + x(T_2) \geq k - |Z|$. W.l.o.g., we suppose that

$$x(T_1) \geq \left\lceil \frac{(k-|Z|)}{2} \right\rceil. \tag{21}$$

Let $\pi' = (V_1, \ldots, V_{i_0-2}, V_{i_0-1} \cup V_{i_0}, V_{i_0+1}, \ldots, V_{p+1})$ be a partition. As π is a Steiner-SP-partition which contains more than three elements, π' is also a Steiner-SP-partition with p elements. Then, by the induction hypothesis, we have the following valid Steiner-SP-partition inequality induced by π'.

$$x(\delta_{G\setminus Z}(V_1, \ldots, V_{i_0-2}, V_{i_0-1} \cup V_{i_0}, V_{i_0+1}, \ldots, V_{p+1})) \geq \left\lceil \frac{k-|Z|}{2} \right\rceil p - 1. \tag{22}$$

By summing the inequalities (21) and (22), we get

$$x(\delta_{G\setminus Z}(V_1, \ldots, V_p, V_{p+1})) \geq \left\lceil \frac{k-|Z|}{2} \right\rceil (p+1) - 1. \tag{23}$$

Hence, we have the result. □

3.6 The rooted partition inequalities

A further class of valid inequalities is the rooted partition inequalities. We consider p demands, $|D| \geq p \geq 2$, of the form (s, t_i), $i = 1, \ldots p$, for $s \in V$ and $t_i \in V \setminus \{s\}$. Let $(V_0, V_1, \ldots, V_p)$ be a partition of V such that $s \in V_0$ and $t_i \in V_i$, for all $i \in \{1, \ldots, p\}$. This partition is called a *rooted partition*. Huygens et al. [25] showed that, for any $L \geq 2$, the following inequality is valid for the 2EDHP polytope.

$$x(\delta(V_0, V_1, \ldots, V_p)) \geq \left\lceil \frac{(L+1)p}{L} \right\rceil. \tag{24}$$

For a subset $Z \subset V$ with $|Z| = k-2$, the following inequality is valid for kNDHP(G, L).

$$x(\delta_{G\setminus Z}(V_0, V_1, \ldots, V_p)) \geq \left\lceil \frac{(L+1)p}{L} \right\rceil. \tag{25}$$

3.7 *st*-jump inequalities

Theorem 7 *Suppose that $|V| \geq 5$ and $L = 3$. Let $(s,t) \in D$, $Z \subset V$, and consider the partition $\pi = (V_0, V_1, \ldots, V_4)$ of $V \setminus Z$ such that $s \in V_0$ and $t \in V_4$. Let U_i be a set of nodes of V_i, $i = 1, 2, 3$, such that $|U_i| = k-1$. Then the st-jump inequality*

$$\sum_{i=0}^{2} x([V_i, V_{i+2}]) + \sum_{i=0}^{1} \sum_{j \geq i+3}^{4} 2x([V_i, V_j]) + \sum_{i=0}^{1} x([V_i, V_{i+1} \setminus U_{i+1}]) + \sum_{i=2}^{3} x([V_i \setminus U_i, V_{i+1}]) \geq \left\lceil \frac{4k+3}{5} \right\rceil \tag{26}$$

is valid for the kNDHP$(G, 3)$.

Proof Let $U_1 \subset V_1$, $U_2 \subset V_2$, and $U_3 \subset V_3$ and let T_1, T_2, T_3, and T_4, be the *L-st-path-cuts* induced by $(V_0, U_1, V_2 \cup V_1 \setminus U_1, V_3, V_4)$, $(V_0, V_1, U_2, V_3 \cup V_2 \setminus U_2, V_4)$, $(V_0, V_1 \cup V_2 \setminus U_2, U_2, V_3, V_4)$, and $(V_0, V_1, V_2 \cup V_3 \setminus U_3, U_3, V_4)$, respectively. Then, by summing the L-st-path-cut inequalities induced by T_i, $i = 1, \ldots, 4$, and the following st-node-cut inequalities induced by V_0 and U_1, $V_0 \cup V_1$ and U_2, and V_4 and U_3,

$$x(\delta_{G\setminus\{U_1\}}(V_0)) \geq k-1,$$
$$x(\delta_{G\setminus\{U_2\}}(V_0 \cup V_1)) \geq k-1,$$
$$x(\delta_{G\setminus\{U_3\}}(V_4)) \geq k-1,$$

we obtain the inequality

$$\sum_{i=1}^{4} x(T_i) + x(\delta_{G\setminus\{U_1\}}(V_0)) + x(\delta_{G\setminus\{U_2\}}(V_0 \cup V_1)) + x(\delta_{G\setminus\{U_3\}}(V_4)) \geq 4k+3.$$

This together with

$$\begin{aligned} x(e) \geq 0, &\quad \text{for all } e \in \delta(V_1 \setminus U_1, V_3) \cup \delta(V_2 \setminus U_2, V_4) \cup \delta(V_1, V_3 \setminus U_3),\\ 3x(e) \geq 0, &\quad \text{for all } e \in \delta(V_0, V_1 \setminus U_1 \cup V_4) \cup \delta(V_1, V_2 \setminus U_2) \cup \delta(V_3 \setminus U_3, V_4),\\ 4x(e) \geq 0, &\quad \text{for all } e \in \delta(V_0 \cup V_2 \setminus U_2 \cup V_3) \cup \delta(V_1, V_4), \end{aligned}$$

gives the inequality

$$\sum_{i=0}^{2} 5x([V_i, V_{i+2}]) + \sum_{i=0}^{1} \sum_{j \geq i+3}^{4} 10x([V_i, V_j]) + \sum_{i=0}^{1} 5x([V_i, V_{i+1} \setminus U_{i+1}]) + \sum_{i=2}^{3} 5x([V_i \setminus \{u_i\}, V_{i+1}]) \geq 4k+3.$$

Dividing the resulting inequality by 5, and rounding up the right-hand side, we obtain inequality (26). □

4 Facets of the kNDHP polytope

In this section, we investigate the conditions under which the inequalities presented in the previous section define facets of kNDHP(G, L). First, we discuss the dimension of kNDHP(G, L).

An edge $e \in E$ is said to be *essential* if there is no solution of the kNDHP on the graph obtained by deleting the edge e from G. Therefore, e is essential if and only if it belongs to either an st-cut or an L-st-path-cut of cardinality k, or, to an st-node-cut or an L-st-path-node-cut of cardinality $k-|Z|$. Then, we have the following theorem.

Theorem 8 *dim(kNDHP(G, L)) $= |E| - |E^*|$, where $|E^*|$ is the set of essential edges.*

Proof We have that the edges of E^* belong to every solution of the problem, meaning that, $x^F(e) = 1$, for all $e \in E^*$, and every solution $F \subseteq E$ of the problem. Then, we have dim(kNDHP$(G, L)) \leq |E| - |E^*|$. By considering the edge sets E and $E_f = E \setminus \{f\}$, for every $f \in E \setminus E^*$, we can clearly see that they form $|E| - |E^*| + 1$ solutions, and their incidence vectors are affinely independent. Therefore, dim (kNDHP$(G, L)) \geq |E| - |E^*|$, and the result follows. □

Corollary 1 *kNDHP(G, L) is full dimensional if $G = (V, E)$ is complete and $|V| \geq k + 2$.*

In the rest of the paper, we assume that G is complete and has at least $k + 2$ nodes. By Corollary 1, kNDHP(G, L) is then full dimensional.

Now, we investigate the conditions under which the trivial and basic inequalities define facets.

Theorem 9 *Inequality $x(e) \leq 1$ defines a facet of kNDHP(G, L) for every $e \in E$.*

Proof For all $f \in E \setminus \{e\}$, consider the edge sets $E_f = E \setminus \{f\}$. Hence, E and the edge sets E_f constitute a set of $|E|$ solutions of the kNDHP. Furthermore, their incidence vectors satisfy $x(e) = 1$ and are affinely independent. □

Theorem 10 *Inequality $x(e) \geq 0$, with $e = uv \in E$, defines a facet of kNDHP(G, L) if one of the following conditions hold.*

1) $|V| \geq k + 3$,
2) $|V| = k + 2$, $|D| \leq k - 1$ *and* $(u, v) \notin D$.

Proof Suppose that $|V| = k + 2$, $|D| \leq k - 1$, and $(u, v) \notin D$. Then, the edge sets $E \setminus \{e\}$ and $E_f = E \setminus \{e, f\}$, for all $f \in E \setminus \{e\}$, are solutions of kNDHP whose incidence vectors satisfy $x(e) = 0$ and are affinely independent.

Now, suppose that $|V| \geq k+3$. Then, for all the demands $(s, t) \in D$, the graph G contains $k + 2$ node-disjoint st-paths (edge st and the $k + 1$ paths of the form (s, u, t), $u \in V \setminus \{s, t\}$). Thus, the sets $E \setminus \{e\}$ and $E_f = E \setminus \{e, f\}$, for all $f \in E \setminus \{e\}$, form a set of $|E|$ solutions of the kNDHP. Moreover, their incidence vectors satisfy $x(e) = 0$ and are affinely independent. Hence, $x(e) \geq 0$ defines a facet of kNDHP(G, L). □

In what follows, we investigate the conditions under which the st-cut and the st-node-cut inequalities define facets of kNDHP(G, L).

Theorem 11 *The st-cut inequalities $x(\delta(W)) \geq k$ define facets of kNDHP(G, L) when $|D| = 1$.*

Proof We denote by $ax \geq \alpha$ the st-cut inequality induced by W, and let $\mathscr{F} = \{x \in k\text{NDHP}(G, L) | ax = \alpha\}$. Suppose there exists a defining facet inequality $bx \geq \beta$ such that $\mathscr{F} \subseteq \mathscr{F}' = \{x \in k\text{NDHP}(G, L) | bx = \beta\}$. We will prove that there is a scalar ρ such that $b = \rho a$. As $|V| \geq k + 2$, there exists $W_1 \subseteq W \setminus \{s\}$ and $W_2 \subseteq \overline{W} \setminus \{t\}$ such that $|W_1| + |W_2| = k$. Let $E_1 = \{sv, v \in W_2\} \cup \{ut, u \in W_1\}$ and $T_1 = E_1 \cup E_0$ where $E_0 = E(W) \cup E(\overline{W})$. Clearly, T_1 is a solution of the kNDHP, and its incidence vector satisfies $ax \geq \alpha$ with equality. Consider an edge $e \in E_1$. It is not hard to see that $T_2 = (T_1 \setminus \{e\}) \cup \{\text{st}\}$ is a solution of the kNDHP and its incidence vector also satisfies $ax \geq \alpha$ with equality. Thus, $bx^{T_1} = bx^{T_2}$. Since $bx^{T_2} = bx^{T_1} - b(e) + b(st)$, we obtain that $b(e) = b(\text{st})$. As e is an arbitrary edge in E_1, this implies that

$$b(e) = b(\text{st}) = \rho \text{for some } \rho \in \mathbb{R} \text{ for all } e \in E_1. \tag{27}$$

Now consider an edge $f = uv \in \delta(W) \setminus E_1$, with $u \in W \setminus \{s\}$ and $v \in \overline{W} \setminus \{t\}$. We distinguish two cases.

Case 1 $u \in W_1$, $v \in W_2$.

Consider $T_3 = (T_1 \setminus \{\text{sv, ut}\}) \cup \{f, \text{st}\}$. Clearly, T_3 is a solution of the kNDHP and its incidence vector satisfies $ax = \alpha$. Hence, we have that $bx^{T_3} = bx^{T_1}$. This implies that $b(\text{sv}) + b(\text{ut}) = b(f) + b(\text{st})$. From Eq. 27, it follows that $b(f) = \rho$.

Case 2 $u \in W_1$ (resp. $u \in W \setminus (W_1 \cup \{s\})$), $v \in \overline{W} \setminus (W_2 \cup \{t\})$ (resp. $v \in W_2$).

Consider the edge set $T_4 = (T_1 \setminus \{\text{tu}\}) \cup \{f\}$. It is easy to see that T_4 is a solution of kNDHP such that $ax^{T_4} = \alpha$. Hence, $bx^{T_4} = \beta$. As $\text{bx}^{T_1} = \beta$, it follows that $b(f) = b(\text{tu}) = \rho$.

If $u \in W \setminus (W_1 \cup \{s\})$ and $v \in W_2$, we also obtain by symmetry that $b(f) = \rho$.

Thus, together with Eq. 27, we obtain that $b(e) = \rho$ for all $e \in \delta(W)$.

Now consider an edge $e \in E_0$, and suppose, w.l.o.g., that $e \in E(W)$. If e does not belong to an L-st-path of T_1, then the edge set $T_5 = T_1 \setminus \{e\}$ also induces a solution of the kNDHP and satisfies $\mathrm{ax}^{T_5} = \alpha$. Hence, we have that $\mathrm{bx}^{T_5} = \mathrm{bx}^{T_1}$ implying $b(e) = 0$. If e belongs to an L-st-path of T_1, say (su, ut) where $e =$ su, then the edge set $T_6 = (T_1 \setminus \{\mathrm{su}, \mathrm{ut}\}) \cup \{\mathrm{st}\}$ induces a solution of the kNDHP, and its incidence vector satisfies $\mathrm{ax}^{T_6} = \alpha$. Consequently, $\mathrm{bx}^{T_7} = \mathrm{bx}^{T_1}$ and therefore, $b(\mathrm{st}) = b(\mathrm{su}) + b(\mathrm{ut})$. As (27), $b(\mathrm{ut}) = b(\mathrm{st})$, it follows that $b(\mathrm{su}) = 0$.

Hence, $b(e) = 0$ for all $e \in E_0$.

Finally, we have that

$$b(e) = \begin{cases} \rho & \text{for all } e \in \delta(W), \\ 0 & \text{if not.} \end{cases}$$

Consequently, $b = \rho a$ with $\rho \in \mathbb{R}$, which finishes the proof. □

Theorem 12 *If $|V| \geq 2k+1$ and $|D| = 1$ with $D = \{(s,t)\}$, then every st-node-cut inequality $x(\delta_{G\setminus Z}(W)) \geq k - |Z|$ where $Z \subset V \setminus \{s,t\}$, and such that $s \in W$, $t \notin W$ and $W \setminus \{s\} \neq \emptyset \neq V \setminus ((W \cup Z) \setminus \{t\})$, defines a facet of kNDHP(G, L).*

Proof Let us denote by $\mathrm{ax} \geq \alpha$ the inequality (3) induced by W and Z, and let $\mathrm{bx} \geq \beta$ be a facet defining inequality of kNDHP(G, L) such that $\{x \in k\text{NDHP}(G, L) : \mathrm{ax} = \alpha\} \subseteq \{x \in k\text{NDHP}(G, L) : \mathrm{bx} = \beta\}$. As before, we will show that there exists a scalar $\rho \in \mathbb{R}$ such that $b = \rho a$.

The idea of the proof is to use the fact that $x(\delta_{G\setminus Z}(W)) \geq k - |Z|$ is a valid cut inequality for $(k - |Z|)$NDHP$(G \setminus Z, L)$, and hence, by Theorem 11, defines a facet of $(k - |Z|)$NDHP$(G \setminus Z, L)$. Thus, there exist dim($(k - |Z|)$NDHP$(G \setminus Z, L)$) solutions of the $(k - |Z|)$NDHP on $G \setminus Z$ whose incidence vectors satisfy $x(\delta_{G\setminus Z}(W)) \geq k - |Z|$ with equality and are affinely independent. In what follows, we will use these solutions to build $|E|$ solutions of the kNDHP on G satisfying $x(\delta_{G\setminus Z}(W)) \geq k - |Z|$ with equality and which are affinely independent. Notice that as G is complete, $|Z| \leq k - 1$ and $|V| \geq 2k + 1$, $G \setminus Z$ is also complete with $|V \setminus Z| \geq k + 2$. Thus, by Corollary 1, the polytope $(k - |Z|)$NDHP$(G \setminus Z, L)$ is full dimensional, and hence dim $((k-|Z|)\text{NDHP}(G \setminus Z, L)) = |E| - |\delta(Z)| - |E(Z)|$.

As $x(\delta_{G\setminus Z}(W)) \geq k - |Z|$ defines a facet of $(k - |Z|)$NDHP$(G \setminus Z, L)$, there must exist $m' = |E| - |\delta(Z)| - |E(Z)|$ solutions of the $(k - |Z|)$NDHP on $G \setminus Z$, denoted by T'_i, $i = 1, \ldots, m'$, whose incidence vectors are affinely independent and satisfy $x(\delta_{G\setminus Z}(W)) = k - |Z|$.

The edge sets $T_i = T'_i \cup \delta(Z) \cup E(Z)$, for all $i \in \{1, \ldots, m'\}$, induce solutions of the kNDHP. Indeed, since G is complete, the paths (s, z, t), $z \in Z$, form a set of $|Z|$ st-paths of length 2 in G. As these st-paths are node-disjoint and do not intersect $V \setminus (Z \cup \{s, t\})$, they form with the s-paths of T'_i a set at least k node-disjoint st-paths in G, for $i = 1, \ldots, m'$. Which implies that T_i, $i = 1, \ldots, m'$ are solutions of kNDHP. Furthermore, their incidence vectors satisfy $x(\delta_{G\setminus Z}(W)) = k - |Z|$ and are affinely independant.

Let a' and b' be the restriction on $E \setminus (\delta(Z) \cup E(Z))$ of a and b, respectively. Thus, we have $a'x^{T_i} = \alpha$, for $i = 1, \ldots, m'$. Therefore, $b'x^{T_i} = \beta$, for $i = 1, \ldots, m'$. As x^{T_i}, $i = 1, \ldots, m'$, are affinely independent and $\alpha \neq 0$, it follows that x^{T_i}, $i = 1, \ldots, m'$, are linearly independent. Consequently, a is the unique solution of the system $a'x^{T_i} = \alpha$, for $i = 1, \ldots, m'$. Let ρ be such that $\beta = \rho\alpha$. It then follows that $b' = \rho a'$. This implies that $b(e) = 0$ for all $e \in E(W) \cup E(\overline{W})$.

Now we will show that $b(e) = 0$ for all $e \in \delta(Z) \cup E(Z)$. Let us denote the edges of $E(Z) \cup \delta(Z) \setminus \bigcup_{z\in Z} \{\mathrm{sz}, \mathrm{zt}\}$ by e_j, $j = 1, \ldots, |\delta(Z)| + |E(Z)| - 2|Z|$. Consider the edge sets $\Gamma_{m'+j} = T_{m'} \setminus \{e_j\}$, for $j = 1, \ldots, |\delta(Z)| + |E(Z)| - 2|Z|$. We can see that these sets induce solutions of the kNDHP, and their incidence vectors satsify $x(\delta_{G\setminus Z}(W)) = k - |Z|$. As $\mathrm{ax}^{T_{m'}} = ax^{\Gamma_{m'+j}} = \alpha$, it follows that $\mathrm{bx}^{T_{m'}} = bx^{\Gamma_{m'+j}} = \beta$. Hence, $b(e_j) = 0$ for $j = 1, \ldots, |\delta(Z)| + |E(Z)| - 2|Z|$.

Let T_1 be the set among $T_1, \ldots, T_{m'}$ containing the edge st. Such a set exists since the inequality defines a facet of kNDHP$(G\setminus, L)$ on $G \setminus Z$ different from a trivial inequality. As $W \setminus \{s\} \neq \emptyset \neq V \setminus ((W \cup Z) \setminus \{t\})$. Let $u_1 \in W \setminus \{s\}$, $u_2 \in (V \setminus (W \cup Z)) \setminus \{t\}$ and $z \in Z$. Consider the edge sets $T_0 = (T_1 \setminus \{\mathrm{sz}\}) \cup \{\mathrm{su}_1, u_1 z\}$ and $T'_0 = (T_1 \setminus \{\mathrm{zt}\}) \cup \{\mathrm{sz}, \mathrm{zu}_2\}$. T_0 and T_1 are solutions of the kNDHP (recall that the path (sz, zt) belongs to T_i). Moreover we have $\mathrm{ax}^{T_0} = \mathrm{ax}^{T_1} = \alpha$. Thus $\mathrm{bx}^{T_1} = \mathrm{bx}^{T_0} = \mathrm{bx}^{T'_0} = \beta$. As $b(\mathrm{su}_1) = b(u_1 z) = b(\mathrm{zu}_2) = b(u_2 t) = 0$, it follows that $b(\mathrm{sz}) = b(\mathrm{zt}) = 0$.

Therefore, $b = \rho a$, which ends the proof of the theorem. □

In what follows, we describe conditions under which the L-st-path and L-node-st path-cut inequalities define facets when $L = 3$.

Theorem 13 *If $|D| = 1$, a 3-st-path inequality (2) induced by a partition $\pi = (V_0, \ldots V_4)$ with $s \in V_0$ and $t \in V_4$, defines a facet of kNDHP$(G, 3)$ if and only if*

(1) $|V_0| = |V_4| = 1$,
(2) $|[s, V_1]| + |[V_3, t]| \geq k$.

Proof Let T be the 3-path-cut induced by $\pi = (V_0, \ldots, V_4)$ such that $s \in V_0$ and $t \in V_4$. Let us denote by $ax \geq \alpha$ the L-st-path inequality induced by T, and let $\mathscr{F} = \{x \in k\text{NDHP}(G, 3)|ax = \alpha\}$.

Necessity (1) We will show that if $|V_0| \geq 2$, inequality $x(T) \geq k$ does not define a facet. The case where $|V_4| \geq 2$ follows by symmetry. Suppose that $|V_0| \geq 2$ and consider the partition $\pi' = (V_0', \ldots, V_4')$ given by

$$V_0' = \{s\},$$
$$V_1' = V_1 \cup (V_0 \setminus \{s\}),$$
$$V_i' = V_i, i = 2, 3, 4.$$

The partition π' produces a 3-path-cut inequality $x(T') \geq k$, where $T' = T \setminus [V_0 \setminus \{s\}, V_2]$. Since G is complete, $[V_0 \setminus \{s\}, V_2] \neq \emptyset$ and T' is strictly contained in T. Thus, $x(T) \geq k$ is redundant with respect to the inequalities

$$x(T') \geq k,$$
$$x(e) \geq 0 \quad \text{for all} \quad e \in [V_0 \setminus \{s\}, V_2],$$

and cannot define a facet.

(2) Suppose that Condition (1) holds, and that $\mathscr{F}$ is a facet of kNDHP $(G, 3)$ different from a trivial inequality. Thus, there exists a solution F of the kNDHP such that $x^F \in \mathscr{F}$ and $F \cap [V_1, V_3] \neq \emptyset$. If this is not the case, then $\mathscr{F}$ would be equivalent to a facet defined by any of the inequalities $x(e) \geq 0$, $e \in [V_1, V_3]$. Note that, since each 3-st-path of F intersects T at least once and $|F \cap T| = k$, F necessarily contains exactly k node-disjoint 3-st-paths. Moreover, each of these paths intersects T only once. If u_i is a node of V_i, $i = 1, \ldots, 3$, this implies that every 3-st-path of F is of the form

(i) $(\text{su}_1, u_1u_2, u_2t)$, $(\text{su}_2, u_2u_3, u_3t)$, (su_1, u_1t), (su_3, u_3t), (st) or
(ii) $(\text{su}_1, u_1u_3, u_3t)$.

If P is one of these st-paths, then $|P \cap A| = 1$ (resp. $|P \cap A| = 2$) if P is of type (i) (resp. (ii)), where $A = [s, V_1] \cup [V_3, t] \cup \{st\}$. As $F \cap [V_1, V_3] \neq \emptyset$, it follows that F contains at least one path of type (ii) and therefore $|F \cap A| \geq k + 1$. Hence $|[s, V_1]| + |[V_3, t]| \geq k$.

Sufficiency Suppose that Conditions (1) and (2) hold. Now suppose that there exist a facet defining inequality $\text{bx} \geq \beta$ such that $\mathscr{F} \subseteq \{x \in k\text{NDHP}(G, 3)|\text{bx} = \beta\}$. As before, we will show that there exists a scalar $\rho \neq 0$ such that $b = \rho a$.

As $|[s, V_1]| + |[V_3, t]| \geq k$, there exist two node sets $U_1 \subseteq V_1$ and $U_3 \subseteq V_3$ such that $|U_1| + |U_3| = k$. Consider the edge subset S_1 formed by the st-paths (su, ut), $u \in U_1 \cup U_3$. Clearly, these st-paths form a set of k node-disjoint 3-st-paths. Moreover, each of these paths intersects T only once. Thus, S_1 induces a solution of kNDHP and its incidence vector belongs to $\mathscr{F}$.

Let $e \in S_1 \cap T$. Let $S_2 = (S_1 \setminus \{e\}) \cup \{st\}$. Since S_2 is a solution of the kNDHP whose incidence vector belongs to $\mathscr{F}$, we have $\text{bs}^{S_2} = \text{bx}^{S_1} = \beta$, implying that $b(e) = b(\text{st})$. As e is an arbitrary edge, we obtain that

$$b(e) = \rho \text{ for all } e \in (S_1 \cap T) \cup \{\text{st}\}, \text{ for some } \rho \in \mathbb{R}. \quad (28)$$

Consider now $e \in E \setminus T$. If $e \notin S_1$, clearly $S_3 = S_1 \cup \{e\}$ is a solution of kNDHP. Moreover, its incidence vector belongs to $\mathscr{F}$. Hence, $b(e) = \text{bx}^{S_3} - \text{bx}^{S_1} = 0$. If $e \in S_1 \setminus T$, then e is either of the form su, $u \in U_1$, or vt, $v \in U_3$. Suppose, w.l.o.g., that $e = \text{su}$, the case where $e = \text{vt}$ is similar. Note that, by the definition of S_1, ut also belongs to S_1. Let $S_4 = (S_1 \setminus \{\text{su}, \text{ut}\}) \cup \{\text{st}\}$. We have that S_4 induces a solution of the kNDHP and $x^{S_4} \in \mathscr{F}$. Hence, $\text{bx}^{S_4} = \text{bx}^{S_1} = \beta$ and, in consequence, $b(\text{su}) + b(\text{ut}) = b(\text{st})$. As, by Eq. 28, $b(ut) = b(st)$, we have that $b(\text{su}) = 0$. Thus, we obtain that

$$b(e) = 0 \text{ for all } e \in E \setminus T. \quad (29)$$

Now let $e \in T \setminus S_1$. Suppose that $e = \text{sv}$ with $v \in V_2$. The case where $e \in [V_2, t]$ is similar. By construction S_1 contains an st-path of the form (su_3, u_3t) where u_3 is a node of V_3. Then the edge set $S_5 = (S_1 \setminus \{\text{su}_3\}) \cup \{e, \text{vu}_3\}$ is a solution of the kNDHP whose incidence vector belongs to $\mathscr{F}$. Thus, $b^{S_5} - b^{S_1} = b(e) + b(\text{vu}_3) - b(\text{su}_3) = 0$. From Eqs. 28 and 29, we then get $b(e) = \rho$.

Let $e = \text{sv}$ with $v \in V_3$. The case where $e \in [V_1, t]$ is similar. Consider the edge set $S_6 = (S_1 \setminus \{\text{su}_3\}) \cup \{e, vt|\}$, where u_3 is a node of U_3, which induces a solution of the kNDHP. Moreover, its incidence vector belongs to $\mathscr{F}$. Hence $bx^{S_6} - bx^{S_1} + b(vt) = b(e) - b(su_3) + b(vt) = 0$. By Eqs. 28 and 29, we get $b(e) = \rho$.

Now suppose that $e = \text{uv} \in [V_1, V_3]$. If $u \in U_1$ and $v \in U_3$, then by considering the edge set $S_8 = (S_1 \setminus \{\text{ut}, \text{sv}\}) \cup \{e, \text{st}\}$, which is a solution of kNDHP with $x^{T_8} \in \mathscr{F}$, we get $b(e) + b(\text{st}) = b(\text{sv}) + b(\text{ut})$. From Eqs. 28 and 29, we have that $b(e) = \rho$. If $u \notin U_1$ and $v \in U_3$, then by considering the edge set $S_9 = (S_1 \setminus \{\text{sv}\}) \cup \{\text{su}, e\}$, we obtain along the same line that $b(e) = \rho$. If $u \in U_1$ and $v \notin U_3$, it follows by symmetry that $b(e) = \rho$. If $u \notin U_1$ and $v \notin U_3$, since the edge set $S_{10} = (S_1 \setminus \{\text{su}_1, u_1t\}) \cup \{\text{su}, e, \text{vt}\}$ is a solution of kNDHP with $x^{T_{10}} \in \mathscr{F}$, we get as before $b(e) = \rho$. Thus, we obtain

$$b(e) = \rho \text{ for all } e \in T \setminus (S_1 \cup \{\text{st}\}). \quad (30)$$

From Eqs. 28–30, we have

$$b(e) = \begin{cases} \rho & \text{for all } e \in T, \\ 0 & \text{if not.} \end{cases}$$

Therefore, $b = \rho a$, and the proof is complete. □

Theorem 14 *If $|D| = 1$, a 3-st-node-path-cut inequality (4) induced by a node subset $Z \subset V$, such that $|Z| \leq k - 1$,* *and a partition $\pi = (V_0, \dots V_4)$ of $V \setminus Z$, with $s \in V_0$ and $t \in V_4$, defines a facet of kNDHP$(G, 3)$ if and only if*

(1) $|V_0| = |V_4| = 1$,
(2) $|[s, V_1]| + |[V_3, t]| \geq k - |Z|$.

Proof The idea of the proof is the same as that used in proving Theorem 12. We can also use the fact that a 3-st-node-path-cut inequality, $x(T_{G\setminus Z}) \geq k - |Z|$, for some 3-st-path-cut T and some node set $Z \subset V\setminus\{s, t\}$, is valid for $(k - |Z|)$NDHP$(G \setminus Z, 3)$ (recall that $(k - |Z|)$NDHP$(G \setminus Z, 3)$ is the polytope associated with the 3-hop-constrained st-path problem on the graph $G \setminus Z$).

Note as before that G is complete, $|Z| \leq k - 1$ and $|V| \geq 2k + 1$, then $G \setminus Z$ is complete with $|V \setminus Z| \geq k + 2$. By Corollary 1, the polytope $(k - Z)$NDHP$(G \setminus Z, 3)$ is full dimensional. Thus, dim $((k - |Z|)$NDHP$(G \setminus Z, 3))=$ $|E| - |\delta(Z)| - |E(Z)|$.

As $x(T_{G\setminus Z}) \geq k - |Z|$ defines a facet of $(k - |Z|)$NDHP$(G\setminus Z, 3)$, there exist $n' = |E|-|\delta(Z)|-|E(Z)|$ solutions of the $(k - |Z|)$NDHP on $G \setminus Z$. We will denote them by S'_i, $i = 1, \dots, n'$, their incidence vectors are affinely independent and satisfy $x(T_{G\setminus Z}) = k - |Z|$. The st-paths of S'_i, $i = 1, \dots, m$, are node-disjoint, hence they are solutions of the polytope $(k - |Z|)$NDHP$(G \setminus Z, 3)$.

The edge sets $S_i = S'_i \cup \delta(Z) \cup E(Z)$, for all $i \in \{1, \dots, n'\}$, induce solutions of the kNDHP. Since S'_i, $i \in \{1, \dots, n'\}$ is a solution of the $(k-|Z|)$NDHP on $G\setminus Z$, there exist $(k - |Z|)$ st-paths of length at most 3, in the subgraph of $G \setminus Z$ induced by S'_i.

We will denote them by $H_l, l = 1, \dots, k-|Z|$. Moreover, as G is complete, the edges sz and zt, for all $z \in Z$, are in G, and the sets (s, z, t), $z \in Z$, form $|Z|$ st-paths of length 2 in G. Hence, the paths H_l, $l = 1, \dots, k - |Z|$ and (s, z, t), $z \in Z$, are node-disjoint. Thus, the sets S_i, $i = 1, \dots, n'$ induce n' solutions of the kNDHP on G. Furthermore, their incidence vectors satisfy $x(T_{G\setminus Z}) = k - |Z|$.

Let a' and b' be the restriction on $E \setminus (\delta(Z) \cup E(Z))$ of a and b, respectively. Thus, we have $a'x^{S_i} = \alpha$, for $i = 1, \dots, n'$. Therefore, $b'x^{S_i} = \beta$, for $i = 1, \dots, n'$. As x^{S_i}, $i = 1, \dots, n'$, are affinely independent and $\alpha \neq 0$, it follows that $x^{S_i} \neq 0$, $i = 1, \dots n'$, and hence, x^{S_i}, $i = 1, \dots, n'$, are linearly independent. Consequently, a is the unique solution of the system $a'x^{S_i} = \alpha$, for $i = 1, \dots, n'$. Let ρ be such that $\beta = \rho\alpha$. It then follows that $b' = \rho a'$. This implies that $b(e) = 0$ for all $e \in E \setminus T$.

Now we will show that $b(e) = 0$ for all $e \in \delta(Z) \cup E(Z)$. Let us denote the edges of $E(Z) \cup \delta(Z) \setminus \bigcup_{z\in Z} \{sz, zt\}$ by e_j, $j = 1, \dots, |\delta(Z)| + |E(Z)| - 2|Z|$. Consider the edge set $\Omega_{n'+j} = S_{n'} \setminus \{e_j\}$, for $j = 1, \dots, |\delta(Z)| + |E(Z)| - 2|Z|$. These sets clearly induce solutions of the kNDHP, and their incidence vectors satsify $x(T_{G\setminus Z}) = k - |Z|$. As $\text{ax}^{S_{n'}} = \text{ax}^{\Omega_{n'+j}} = \alpha$, it follows that $\text{bx}^{S_{n'}} = \text{bx}^{\Omega_{n'+j}} = \beta$. Hence, $b(e_j) = 0$ for $j = 1, \dots, |\delta(Z)| + |E(Z)| - 2|Z|$.

Let S_1 be the set among $S_1, \dots, S_{n'}$ containing the edge st. Such a set exists since, as noted before, $x^{S_i} \neq 0$ for $i = 1, \dots, n'$. Let $u_1 \in V_1$, $u_2 \in V_L$ and $z \in Z$. Consider the edge set $S_0 = (S_1 \setminus \{\text{sz}\}) \cup \{\text{su}_1, u_1z\}$ and $S'_0 = (S_1 \setminus \{\text{zt}\}) \cup \{\text{sz}, \text{zu}_2\}$. It clearly induces a solution of the kNDHP. Moreover, we have $x(T_{G\setminus Z}) = k - |Z|$. Thus, $\text{bx}^{T_1} = \text{bx}^{T_0} = \text{bx}^{T'_0} = \beta$. As $b(\text{su}_1) = b(u_1z) = b(\text{zu}_2) = b(u_2t) = 0$, it follows that $b(\text{sz}) = b(\text{zt}) = 0$.

Therefore $b = \rho a$, which ends the proof of the theorem. □

Note that Theorems 9, 10, 11, and 12 are valid for $L \geq 4$.

Lemma 2 *The double cut inequality induced by the node sets $V_0^1, V_0^2 \cup V_1, V_2, \dots, V_{L+1}$ of $V \setminus Z$, $F \subseteq E$ and $\{s, t\} \in D$ with $s \in V_0^1$ and $t \in V_{L+1}$, can be written as*

$$x(T_{G\setminus Z}) + x(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) + x(\delta_{G\setminus Z}(V_1)) + x(\overline{E} \setminus F) - x(F) + |F| \geq 3(k - |Z|) + 1 \quad (31)$$

where $T_{G\setminus Z}$ is the L-st-node-path-cut induced by the partition $(V_0^1, V_0^2 \cup V_1, V_2, \dots, V_{L+1})$. Moreover, the double cut inequality (13) is tight for a solution $\tilde{x} \in \mathbb{R}^E$ if and only if one of the following conditions holds.

i) $\tilde{x}(\overline{E} \setminus F) - \tilde{x}(F) + |F| = 1$ *and* $\tilde{x}(T_{G\setminus Z}) = \tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) = \tilde{x}(\delta_{G\setminus Z}(V_1)) = k - |Z|$*;*
ii) $\tilde{x}(\overline{E} \setminus F) - \tilde{x}(F) + |F| = 0$ *and*
   a) $\tilde{x}(T_{G\setminus Z}) = k-|Z|+1$, $\tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) = k-|Z|$ *and* $\tilde{x}(\delta_{G\setminus Z}(V_1)) = k - |Z|$*;*
   b) $\tilde{x}(T_{G\setminus Z}) = k-|Z|$, $\tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) = k-|Z|+1$ *and* $\tilde{x}(\delta_{G\setminus Z}(V_1)) = k - |Z|$*;*
   c) $\tilde{x}(T_{G\setminus Z}) = k - |Z|$, $\tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) = k - |Z|$ *and* $\tilde{x}(\delta_{G\setminus Z}(V_1)) = k - |Z| + 1$*;*

Proof Let C be the double cut inducing inequality (13). Then, inequality (13) can be written as

$$x(C \setminus \overline{E}) + x(\overline{E} \setminus F) \geq \frac{3(k - |Z|) - |F| + 1}{2}.$$

Thus, we have

$$2x(C \setminus \overline{E}) + 2x(\overline{E}) - 2x(F) \geq 3(k - |Z|) - |F| + 1. \quad (32)$$

By summing the left-hand side of the L-st-node-path-cut inequality induced by $T_{G\setminus Z}$ and the node-cut inequalities induced by $\delta_{G\setminus Z}(V_0^1 \cup V_0^2)$ and $\delta_{G\setminus Z}(V_1)$, we obtain

$$x(T_{G\setminus Z}) + x(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) + x(\delta_{G\setminus Z}(V_1)) = 2x(C \setminus \overline{E}) + x(\overline{E}). \quad (33)$$

By combining Eqs. 32 and 33, we get

$$x(T_{G\setminus Z}) + x(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) + x(\delta_{G\setminus Z}(V_1)) + x(\overline{E}) - 2x(F) \geq 3(k - |Z|) - |F| + 1.$$

Therefore,

$$x(T_{G\setminus Z}) + x(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) + x(\delta_{G\setminus Z}(V_1)) + x(\overline{E} \setminus F) - x(F) + |F| \geq 3(k - |Z|) + 1.$$

Hence, the double cut inequality (13) is equivalent to Eq. 31.

Suppose that the double cut inequality is tight for a solution $\tilde{x}$, that is

$$\tilde{x}(T_{G\setminus Z}) + \tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) + \tilde{x}(\delta_{G\setminus Z}(V_1)) + \tilde{x}(\overline{E} \setminus F - \tilde{x}(F) + |F| = 3(k - |Z|) + 1$$

As $\tilde{x}(T_{G\setminus Z}) \geq k - |Z|$, $\tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) \geq k - |Z|$ and $\tilde{x}(\delta_{G\setminus Z}(V_1)) \geq k - |Z|$, we have that $\tilde{x}(\overline{E} \setminus F) - \tilde{x}(F) + |F| \leq 1$. Thus, if $\tilde{x}(\overline{E} \setminus F) - \tilde{x}(F) + |F| = 1$, we have that $\tilde{x}(T_{G\setminus Z}) = \tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) = \tilde{x}(\delta_{G\setminus Z}(V_1)) = k - |Z|$. If $\tilde{x}(\overline{E} \setminus F) - \tilde{x}(F) + |F| = 0$, then either $\tilde{x}(T_{G\setminus Z})$ or $\tilde{x}(\delta_{G\setminus Z}(V_0^1 \cup V_0^2))$ or $\tilde{x}(\delta_{G\setminus Z}(V_1))$ is equal to $k - |Z| + 1$ and the others are equal to $k - |Z|$, and the statement follows. □

Theorem 15 *The double cut inequality (13) defines a facet of kNDHP$(G, 3)$ only if*

i) $|V_0^1| = |V_4| = 1$,
ii) $|[V_0^1, V_0^2 \cup V_1] \cup [V_3, V_4] \cup [V_0^1, V_4]| \geq k - |Z|$.

Proof i) Let C be the double cut-inducing inequality (13). Using the following family of sets $\Pi = (V_0^1, V_0^2, V_1, V_2, \ldots, V_4)$. Suppose that $|V_0^1| > 1$, the case when $|V_{L+1}| > 1$ is similar. Consider the family of sets $\Pi' = \{\{s\}, V_0^1 \setminus \{s\}, V_0^2, V_1, \ldots, V_4\}$. Let C' be the double cut induced by Π' and F. Since $C = C' \cup [V_0^1 \setminus \{s\}, V_1]$, then the double cut inequality induced by Π is redundant with respect to the one induced by Π', and the trivial inequalities $x(e) \geq 0$ for all $e \in [V_0^1 \setminus \{s\}, V_1]$. Thus, it does not define a facet.

(ii) Let $\mathscr{F}$ be a facet defining double cut inequality and let $T_{G\setminus Z}$ be the 3-st-node-path-cut induced by the partition $(V_0^1, V_0^2 \cup V_1, V_2, \ldots, V_4)$. As $\mathscr{F}$ defines a facet different from the node-cut inequalities, there exists a solution $x_0 \in \mathscr{F}$ such that $x_0(\delta_{G\setminus Z}(V_0^1 \cup V_0^2)) \geq k - |Z| + 1$. Then by Lemma 2, $x_0(T) = k - |Z|$. Thus, x_0 induces a graph which contains exactly $k - |Z|$ node-disjoint 3-st-paths, $P_1, \ldots, P_{k-|Z|}$. Furthermore, each P_i, $i = 1, \ldots, k - |Z|$ intersects $T_{G\setminus Z}$ in only one edge. Thus, either $P_i \cap [V_0, V_4] \neq \emptyset$ or P_i uses at least one edge between two non-consecutive set of the partition $(V_0^1, V_0^2, V_1, V_2, \ldots, V_4)$. In the latter case, P_i must intersect either $[V_0^1, V_0^2 \cup V_1]$ or $[V_3, V_4]$ or both. Hence, we have that $|[V_0^1, V_0^2 \cup V_1] \cup [V_3, V_4] \cup [V_0^1, V_4]| \geq k - |Z|$. Which ends the proof. □

5 Branch-and-cut algorithm for the *k*NDHP with $L = 3$ and $k \geq 3$

In this section, we present a branch-and-cut algorithm for the *k*NDHP when $L = 3$. First, we present the general framework of the algorithm and then present the separation procedures we have devised for the inequalities involved in the algorithm.

5.1 The general framework

Our algorithm starts by solving the linear relaxation of Formulation (7), that is,

$$\min\{cx | x \in \mathbb{R}_+^E \text{ satisfies } (1) - (6)\}. \tag{34}$$

Since inequalities (1), (2), (3), and (4) are exponential in number in (34), we solve this linear relaxation using the so-called *cutting plane method*. We recall that the cutting plane method finds an optimal solution of a linear program by solving a series of LPs, each of them containing a subset of the constraints of the original LP. For our purpose, the algorithm starts with an LP containing the cut constraints (1) induced by terminal nodes and the trivial inequalities (5) and (6)

$$\begin{aligned} &\text{Min} \sum_{e \in E} c(e)x(e) \\ &\text{s.t.} \\ &x(\delta(u)) \geq k, \text{ for all } u \in R_D, \\ &x(e) \geq 0, \text{ for all } e \in E, \\ &x(e) \leq 1, \text{ for all } e \in E. \end{aligned}$$

Then, it iteratively adds the inequalities (1)–(4) that are violated by the solution x^* of the current LP. The cutting plane algorithm stops when all the inequalities (1)–(4) are satisfied by x^*. In this case, x^* is optimal for Eq. 34). For finding inequalities (1)–(4) that are violated by x^*, if there is any, we solve the so-called separation problem associated with these inequalities. Recall that *the separation problem* associated with a family of inequalities $\mathcal{F}$ and a solution $\overline{x}$ is to verify if $\overline{x}$ satisfies all the inequalities of $\mathcal{F}$, and if not, to exhibit at least one of them which is violated by $\overline{x}$. An algorithm solving a separation problem is called a *separation algorithm*.

At the end of the cutting plane algorithm, if x^* is integral, then it is optimal for the problem (7). If x^* is fractional, then we reinforce the linear relaxation of the problem by adding, if possible, further valid inequalities. For this, we also add the Steiner SP-partition inequalities (19), the double cut inequalities (11) and the Steiner partition inequalities (17) in the cutting plane algorithm. The separation of the inequalities used in the branch-and-cut algorithm are performed in the following order

1. st-cut and L-st-path-cut inequalities,
2. st-node-cut and L-st-node-path-cut inequalities (only for integral solutions),
3. Steiner SP-partition inequalities,
4. double cut inequalities,
5. Steiner partition inequalities.

Notice that the st-node-cut and L-st-node-path-cut inequalities are separated only for integral solutions. Indeed, as we will see in the next subsection, these two families of inequalities can be efficiently separated when the solution x^* is integral.

All the inequalities that are added during the branch-and-cut algorithm are considered as global (i.e., valid at every node of the branch-and-cut tree), and we may add several inequalities at each iteration. Furthermore, we proceed to the separation of a class of inequalities only when the separation of the previous class of inequalities has not found any violated inequalities.

In the following, we describe the separation algorithms we have devised for the inequalities (1)–(4), the Steiner SP-partition inequalities (19), the double cut inequalities (11), and the Steiner partition inequalities (17).

5.2 Separation procedures

5.2.1 Separation of st-cut and 3-st-path-cut inequalities

We discuss first the separation of the st-cut and 3-st-path-cut inequalities (1) and (2). We give the theorem below which shows that the separation problem of these inequalities reduces to computing a maximum flow in a special graph, and hence can be solved in polynomial time.

Theorem 16 *The separation problem of st-cut and 3-st-path-cut inequalities* (1) *and* (2) *reduces to computing maximum flows in a special graph and can be solved in* $O(|D||E|^2|V|)$ *time.*

Proof Let $\overline{x} \in \mathbb{R}^E$ be the solution for which we are separating the natural inequalities (1) and (2). To separate them, we consider the following graph transformation from [2] (see also [13]). Let $(s,t) \in D$ and let $V_{\mathrm{st}} = V \setminus \{s,t\}$, V'_{st} be a copy of V_{st} and $\widetilde{V}_{\mathrm{st}} = V_{\mathrm{st}} \cup V'_{\mathrm{st}} \cup \{s,t\}$. The copy in V'_{st} of a node $u \in V_{\mathrm{st}}$ will be denoted by u'. From G and (s,t), we build the directed graph $\widetilde{G}_{\mathrm{st}} = (\widetilde{V}_{\mathrm{st}}, \widetilde{A}_{\mathrm{st}})$. Its arc set $\widetilde{A}_{\mathrm{st}}$ is obtained as follows. For an edge of the form st $\in E$, we add an arc (s,t) in $\widetilde{A}_{\mathrm{st}}$. For each edge su $\in E$, $u \neq t$, (resp. vt $\in E$, $v \neq s$), we add in $\widetilde{A}_{\mathrm{st}}$ an arc (s,u), $u \in V_{\mathrm{st}}$ (resp. (v',t), $v' \in V'_{\mathrm{st}}$). For each edge uv $\in E$, with $u,v \notin \{s,t\}$, we add two arcs (u,v') and (v,u') in $\widetilde{A}_{\mathrm{st}}$, with $u,v \in V_{\mathrm{st}}$ and $u',v' \in V'_{\mathrm{st}}$. Finally, for each node $u \in V \setminus \{s,t\}$, we add an arc (u,u') in $\widetilde{A}_{\mathrm{st}}$ (see Fig. 7 for an illustration).

Notice that for each $(s,t) \in D$, $|\widetilde{V}_{\mathrm{st}}| = 2|V| - 2$ and $|\widetilde{A}_{st}| = 2|E| - |\delta(s)| - |\delta(t)| + |[s,t]|$.

Bendali et al. [2] showed that there is a one-to-one correspondence between the st-cuts and the 3-st-path-cuts in G and the st-dicuts in $\widetilde{G}_{\mathrm{st}}$ which do not contain arcs of the form (u,u'), for all $u \in V \setminus \{s,t\}$. Moreover, if each arc $a \in \widetilde{A}_{\mathrm{st}}$, corresponding to an edge $e \in E$, is assigned the capacity $\widetilde{c}(a) = \overline{x}(e)$ and each arc of the form (u,u') is assigned an infinite capacity, then the weight of an st-cut or 3-st-path-cut in G with respect to $\overline{x}$ is the same as that of the corresponding st-dicut in $\widetilde{G}_{\mathrm{st}}$ with respect to capacity vector $\widetilde{c}$. Thus, for a given $(s,t) \in D$, there is an st-cut or 3-st-path-cut inequality violated by $\overline{x}$ if and only if there is an st-dicut in $\widetilde{G}_{\mathrm{st}}$ whose capacity is $< k$. Moreover, if there is a violated st-cut or 3-st-path-cut inequality induced by an edge set $C \subseteq E$, that is there is an st-dicut $\widetilde{C} \subseteq \widetilde{A}_{\mathrm{st}}$ whose weight is $< k$, then the edges of C are those corresponding to the arcs of $\widetilde{C}$. Therefore, the separation problem of the st-cut and the 3-st-path-cut inequalities reduces to computing a minimum st-dicut in $\widetilde{G}_{\mathrm{st}}$ with respect to the capacity vector $\widetilde{c}$. By the max-flow min-cut theorem, this can be done by computing a maximum flow from s to t in $\widetilde{G}_{\mathrm{st}}$.

Finally, the maximum flow computation in $\widetilde{G}_{\mathrm{st}}$ can be handled by the Edmonds-Karp algorithm [16] which runs in $O(|\widetilde{A}_{\mathrm{st}}|^2|\widetilde{V}_{\mathrm{st}}|) = O(|E|^2|V|)$ time. Since this procedure is performed $|D|$ times (one for each demand), the whole separation algorithm can be implemented to run in $O(|D||E|^2|V|)$ time, and hence is polynomial. □

Our separation algorithm for st-cut and 3-st-path-cut inequalities is based on Theorem 16. It starts, for each demand $(s,t) \in D$, by building the graph $\widetilde{G}_{\mathrm{st}}$ and then, computing a minimum weight st-dicut, say $\widetilde{C}$, w.r.t. weight vector $\widetilde{c}$. If the weight of such a st-dicut is $< k$, then the edge set C of G corresponding to the arcs of $\widetilde{C}$ corresponds to either a st-cut or a 3-st-path-cut which induces a violated inequality. The separation algorithm stops when it finds, for a given demand, a violated inequality or when all the demands have been considered without finding any violated inequality. From Theorem 16, this algorithm solves the separation problem of inequalities (1) and (2) in polynomial time.

5.2.2 Separation of st-node-cut and 3-st-node-path-cut inequalities

Now, we discuss the separation problem of st-node-cut and 3-st-node-path-cut inequalities (3) and (4). We also assume

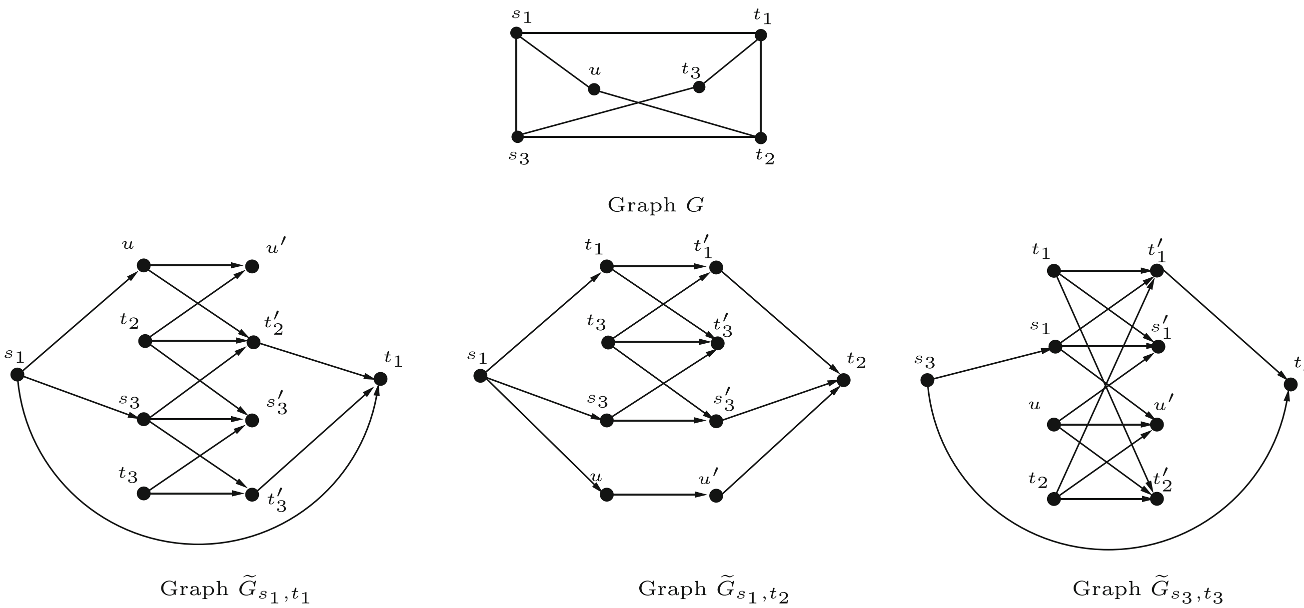

Fig. 7 Construction of graphs $\widetilde{G}_{\mathrm{st}}$ with $D = \{(s_1, t_1), (s_1, t_2), (s_3, t_3)\}$

that the solution $\overline{x}$ is integral and satisfies all the st-cut and 3-st-path-cut inequalities (1) and (2). From Theorem 1, the separation problem of inequalities (3) and (4), for a demand $(s, t) \in D$, reduces to check if there is a node set $Z \subseteq V \setminus \{s, t\}$ and an st-cut $\widetilde{C}$ of $\widetilde{G}_{st} \setminus \widetilde{Z}$ such that $|Z| \leq k-1$ and $\widetilde{y}(\widetilde{C}) < k - |Z|$. Moreover, the computation of both Z and $\widetilde{C}$ can be done after the application of procedure BuildZ (see the proof of Theorem 1 for more details). Finally, notice that procedure BuildZ reduces to compute at most k maximum flows in auxililliary graphs $\widetilde{G}_{st} \setminus \widetilde{Z}$.

Now we describe our separation algorithm for st-node-cut and 3-st-node-path-cut inequalities when x is integral and satisfies all the st-cut and 3-st-path-cut inequalities. For each demand $(s, t) \in D$, we build the graph $\widetilde{G}_{st}$ and let $\widetilde{c}$ be the associated weight vector. Then we build, using Procedure BuildZ, the node set Z and let f be the weight of a minimum weight cut of $\widetilde{G}_{st} \setminus \widetilde{Z}$, w.r.t. weight vector $\widetilde{c}$. If $|Z| \leq k - 1$ and $f < k - |Z|$, then, by Theorem 1, there is an st-node-cut or a 3-st-node-path-cut $C \subseteq E$ which induces an inequality (3) or (4) violated by $\overline{x}$. If $|Z| \geq k$ or $f \geq k - |Z|$, then we move to another demand. The algorithm stops when it has found a violated inequality (3) or (4) for some demand $(s, t) \in D$ or when all the demands have been explored without finding any violated inequality.

If we use Edmonds-Karp algorithm for each maximum flow computation, then the separation algorithm can be implemented to run in $O(|D|k|E|^2|V|)$ time, which is polynomial.

5.2.3 Separation of double cut, Steiner SP-partition, and partition inequalities

Now, we consider the separation of inequalities (11), (19) and (17). For our purpose, we look for those inequalities (11), (19), and (17) defined with a node set $Z = \emptyset$. To separate them, we use the separation heuristics developed in [11].

The heuristic developed for the double cut inequalities is implemented to run in $O\left(|V|^3 \log |V| \frac{(2|V|+|D_{\mathrm{source}}|+|D_{\mathrm{dest}}|)^2}{(|V|-1)(|V|+|D_{\mathrm{source}}|+|D_{\mathrm{dest}}|)}\right)$ time. Here D_{source} and D_{dest} denote the sets of nodes which are, respectively, the source, and destination in a demand, which is polynomial.

For SP-partition inequalities (19), the heuristic proposed by [11] is implemented to run in $O(|V||E|+|D|)$, while the separation heuristic for partition inequalities (17) proposed by [11] is implemented to run in $O(|V||E| + |R|^2(|E| + |D|))$, where R is the set of terminal nodes.

Clearly, the three heuristics run in polynomial time.

5.3 Computational results

We have implemented our branch-and-cut algorithm in C++, using CPLEX 12.5 and concert technology [10]. It was tested on a Xeon Quad-Core E5507 machine with a 2.27 GHz processor and 8GB RAM, running under Linux. The maximum CPU time has been fixed to 5 h. Each instance is composed of a graph from TSPLIB [34] and a set of demands. TSPLIB graphs are complete Euclidean graphs, that is each node is assigned coordinates in the plane, and the weight of each edge is given by the Euclidean distance between its endnodes. The demands used in the instances are randomly generated. Each set of demands is either rooted, that is, of the form $\{(s, t_i) : i = 1, \ldots, d\}$ (s is the root node of the demands), or arbitrary.

The computational results are given in Tables 3, 4, 5, 6, 7, and 8. Each instance is described by the number of nodes of the graph and the number of demands. The number of nodes is preceded either by "r" if the demands are rooted or "a" if they are not rooted. The entries of the various tables presented below are:

- $|V|$: the number of nodes of the graph,
- $|D|$: the number of demands,
- C-LPC : the number of generated st-cut and 3-st-path-cut inequalities,
- NC-NLPC : the number of generated st-node-cut and 3-st-node-path-cut inequalities,
- SP : the number of generated Steiner SP-partition inequalities,
- DC : the number of generated double cut inequalities,
- DC : the number of generated Steiner partition inequalities,
- COpt : value of the best upper bound obtained,
- Gap : the relative error between the best upper bound and the lower bound obtained at the root node of the branch-and-cut tree,
- NSub : the number of nodes in the branch-and-cut tree,
- CPU : total CPU time of the first run in hours:min.sec.

Note that for some instances, the algorithm spends all the CPU time (5 h) without finding any feasible solution. In this case, the best upper bound (COpt) and the error with the lower bound achieved at the root node of the Branch-and-Cut tree (Gap) are indicated with "-".

Our first series of experiments concerns the kNDHP with $k = 3$ and $L = 3$. The results are given in Tables 3 and 4.

We can see that for the rooted instances (Table 3), the algorithm has solved to optimality 4 instances out of 13, with graphs having up to 30 nodes and with 15 demands, and the CPU time varying from 4 to 18 mins. The gap achieved between the best upper bound (that is, the optimal solution) and the lower bound at the root node of the branch-and-cut tree (gap) is relatively small (less than 10%) for these instances. For the instances that have not been solved to optimality, the value of the gap are also relatively small: less than 10% for 5 of them and less 31% for the 4 other instances. Table 3 also shows that a very large number of st-cut and 3-st-path-cut inequalities have been generated during the resolution. Also, a large number of st-node-cut and 3-st-node-path-cut inequalities have been generated for all the instances. We can also see that several Steiner SP-partition have been generated but no double cut and Steiner partition inequalities have been generated.

For the arbitrary demands, the algorithm has solved to optimality only one instance (d-21-11) over nine and has spent all the CPU for the other instances. Also, it has not found even a feasible solution for 5 instances. For the instances r-21-10, d-30-10 and d-30-15, that have not been solved to optimality, the gap between the lower bound at the root node of the branch-and-cut tree and the best upper bound is less than 12%. We can also see, as for the rooted instances, that a very large number of st-cut and 3-st-path-cut inequalities have been generated, but less st-node-cut and 3-st-node-path-cut inequalities have been generated.

Table 3 Results for $k = 3$, $L = 3$ and rooted demands

$|V|$	$|D|$	C-LPC	NC-NLPC	SP	DC	P	COpt	Gap	NSub	CPU
r 21	15	11775	74	10	0	0	5526	9.23	2265	00:04:46
r 21	17	22356	228	0	0	0	5939	9.4	4518	00:18:24
r 21	20	71354	116	0	0	0	6466	9.54	17673	03:10:39
r 30	15	15599	264	14	0	0	10109	6.87	1521	00:12:06
r 30	20	58659	1516	12	0	0	11376	8.41	15280	05:00:00
r 30	25	80999	615	18	0	0	12661	12.33	14281	05:00:00
r 48	20	51038	1632	26	0	0	18337	18.1	6133	05:00:00
r 48	30	66277	898	10	0	0	25437	28.68	5305	05:00:00
r 48	40	69242	257	2	0	0	31693	30.17	5628	05:00:00
r 52	20	49717	1674	22	0	0	11170	9.15	5707	05:00:00
r 52	30	62698	1692	18	0	0	14626	17.11	3845	05:00:00
r 52	40	68794	1024	16	0	0	17920	21.86	4953	05:00:00
r 52	50	77808	142	0	0	0	20873	24.49	4397	05:00:00

Table 4 Results for $k = 3$, $L = 3$ and arbitrary demands

$\|V\|$	$\|D\|$	C-LPC	NC-NLPC	SP	DC	P	COpt	Gap	NSub	CPU
a 21	10	56593	2	0	483	0	6680	8.66	9191	05:00:00
a 21	11	29325	2	0	375	1	6770	6.8	2614	00:57:32
a 30	10	44057	25	18	38	0	10354	6.64	13274	05:00:00
a 30	15	53545	86	0	462	0	13936	11.69	6399	05:00:00
a 48	15	34047	0	0	20	0	–	–	1119	05:00:00
a 48	20	28329	0	2	10	0	–	–	229	05:00:00
a 48	24	23975	0	0	11	0	–	–	103	05:00:00
a 52	20	30157	108	6	41	0	–	–	1735	05:00:00
a 52	26	24217	0	0	96	0	–	–	307	05:00:00

We can also notice that some Steiner SP-partition and a quite large number of double cut inequalities have been generated in the resolution.

Our next series of experiments concerns the kNDHP with $k = 4$ and $L = 3$. The results are given in Tables 5 and 6. Notice that in this case, the Steiner SP-partition and Steiner partition inequalities are not included in the branch-and-cut algorithm as they are redundant w.r.t. st-cut inequalities. Thus, the corresponding columns in Tables 5 and 6 are omitted.

The results of Table 5 show that for the rooted instances, 4 instances over 13 have been solved to optimality. For the other instances, the gap is less than 9% for three instances and less than 22% for six instances. The results also show that a very large number of st-cut and 3-st-path-cut inequalities are generated while a large number of st-node-cut and 3-st-node-path-cut inequalities are generated for all the instances. Also, no double cut inequalities are generated for all the instances we have considered.

For arbitrary demands (Table 6), all the instances have not been solved to optimality within the CPU time limit. Also, for five instances (from d-48-15 to d-52-26) over nine, the algorithm has not found a feasible solution. For the others, the gap is less than 13%. Contrarily to rooted demands, a quite large number of double cut inequalities have been generated.

Now we turn out attention to the resolution of the kNDHP with $k = 5$ and $L = 3$. The results are given in Tables 7 and 8 below.

We can see from Table 7, that for the rooted instances, the algorithm has solved to optimality three instances over 13. For the other ten instances, the gaps is less than 10%, for only three of them. For the remaining instances, the gaps are between 10 and 35%. Also, we notice that a large number of st-cut, 3-st-path-cut, st-node-cut and 3-st-node-path-cut inequalities are generated. However, no Steiner SP-partition, double cut and Steiner partition inequalities are generated. For the arbitrary demands (Table 8), all the instances have not been solved to optimality, and, for four instances, the algorithm has not found a feasible solution. We also notice that few Steiner SP-partition and Steiner

Table 5 Results for $k = 4$, $L = 3$ and rooted demands

$\|V\|$	$\|D\|$	C-LPC	NC-NLPC	DC	COpt	Gap	NSub	CPU
r 21	15	4923	52	0	7322	4.57	1078	00:00:50
r 21	17	5732	24	0	7826	4.56	1186	00:01:06
r 21	20	35317	9	0	8556	5.32	17991	01:11:08
r 30	15	48473	2266	0	14315	6.66	10718	05:00:00
r 30	20	23784	0	0	15041	4.19	5664	00:35:22
r 30	25	54445	595	0	16379	5.93	11631	05:00:00
r 48	20	40090	1784	0	26131	20.86	6929	05:00:00
r 48	30	43988	621	0	29806	16.86	4140	05:00:00
r 48	40	51107	232	0	40037	24.77	4302	05:00:00
r 52	20	39125	1346	0	15480	8.72	5106	05:00:00
r 52	30	42750	2760	0	20976	20.28	5192	05:00:00
r 52	40	49499	831	0	24343	21.52	4865	05:00:00
r 52	50	56313	282	0	26541	17.92	4472	05:00:00

Table 6 Results for $k = 4$, $L = 3$ and arbitrary demands

$\|V\|$	$\|D\|$	C-LPC	NC-NLPC	DC	COpt	Gap	NSub	CPU
a 21	10	50711	108	858	9339	10.37	9674	05:00:00
a 21	11	55432	127	703	9864	12.52	10221	05:00:00
a 30	10	36595	116	152	14582	6.3	9817	05:00:00
a 30	15	39442	37	319	18961	10.19	4593	05:00:00
a 48	15	24750	0	20	–	–	589	05:00:00
a 48	20	20007	0	4	–	–	137	05:00:00
a 48	24	16095	0	2	–	–	47	05:00:00
a 52	20	21556	0	54	–	–	867	05:00:00
a 52	26	14635	0	35	–	–	215	05:00:00

partition inequalities and a quite large number of double cut inequalities have been generated.

In these experiments, we have also tried to check the impact of the different classes of inequalities we have considered in our algorithm. As we can see in the various tables, Steiner SP-partition and double cut inequalities are generated in quite large number, and very few Steiner partition inequalities are found. We also observe that in the three cases $k = 3, 4, 5$, the double cut inequalities are not generated when the demands are rooted, and several of them are generated when the demands are arbitrary. In contrast with double cut inequalities, Steiner SP-partition inequalities are mainly generated when the demands are rooted, and few of them are generated for arbitrary demands. This observation can be compared with those of Diarrassouba et al. [12] who devised a branch-and-cut algorithm for the kNDHP with $k = 2$. In their experiments, they showed that the double cut inequalities were mainly generated when the demands are arbitrary. This suggests that the double cut inequalities (11) are mainly involved in the resolution of the problem when the demands are arbitrary, and when the demands are rooted, Steiner SP-partition inequalities may play an important role in solving the problem.

To conclude this experimental study, we have checked the impact of the connectivity on the resolution of the problem. Such a comparison has been made by Bendali et al. [3], for the k-edge-connected subgraph problem, and by Diarrassouba et al. [13], for the kEHDP, that is the hop-constrained survivable network design problem in which the L-st-paths are required to be edge-disjoint, for each demand $(s, t) \in D$. In both studies, the computational results suggest that the problem becomes easier to solve when the connectivity increases. However, for the kNDHP, our computational results do not allow to make the same conclusion. Indeed, by comparing Tables 3, 8 and 7, we can see that most of the instances that have not been solved to optimality for $k = 3$ have also not been solved to optimality for $k = 4$ and $k = 5$. Also, the number of nodes in the branch-and-cut tree is quite large in the three cases. Also, the different gaps achieved do not allow to see if the problem becomes easier when k increases. In fact,

Table 7 Results for $k = 5$, $L = 3$ and rooted demands

$\|V\|$	$\|D\|$	C-LPC	NC-NLPC	SP	DC	P	COpt	Gap	NSub	CPU
r 21	15	8854	173	0	0	0	9560	3.24	3283	00:03:45
r 21	17	23490	804	0	0	0	10235	3.93	19537	00:54:09
r 21	20	31742	958	0	0	0	11095	4.1	59210	03:18:34
r 30	15	80055	6007	0	0	0	19624	10.01	25045	05:00:00
r 30	20	87973	838	0	0	0	20444	5.78	19283	05:00:00
r 30	25	77565	670	0	0	0	21604	5.31	23220	05:00:00
r 48	20	55964	4334	0	0	0	32753	18.53	11308	05:00:00
r 48	30	59897	1414	0	0	0	41200	22.3	9979	05:00:00
r 48	40	68991	319	0	0	0	48194	20.02	7758	05:00:00
r 52	20	55456	5330	0	0	0	28222	34.95	10387	05:00:00
r 52	30	56528	3283	0	0	0	31443	31.67	9388	05:00:00
r 52	40	61465	1330	0	0	0	30645	20.24	7997	05:00:00
r 52	50	77724	275	0	0	0	33994	17.27	8225	05:00:00

Table 8 Results for $k = 5$, $L = 3$ and arbitrary demands

$\|V\|$	$\|D\|$	C-LPC	NC-NLPC	SP	DC	P	COpt	Gap	NSub	CPU
a 21	10	71158	279	0	645	0	11703	7.8	23196	05:00:00
a 21	11	74831	516	0	1097	1	12533	11.58	24980	05:00:00
a 30	10	49032	305	4	0	0	18613	2.94	20559	05:00:00
a 30	15	56285	242	0	589	0	24043	8.29	9227	05:00:00
a 48	15	37209	91	0	8	0	–	–	2479	05:00:00
a 48	20	28574	0	0	2	0	–	–	303	05:00:00
a 48	24	24246	0	0	0	0	–	–	113	05:00:00
a 52	20	33492	29	0	0	0	30754	17.31	2065	05:00:00
a 52	26	25465	0	0	20	0	–	–	347	05:00:00

for some instances, like r-21-20, the gap decreases as k increases, while for some other instances, like r-30-15, the gap is better when $k = 4$ than when $k = 3$ and $k = 5$. Even, for some instances, like r-48-20, the gaps increase as k increases. The observations are the same for the arbitrary demands, that is, we cannot conclude from Tables 4, 6 and 8 that the resolution of the kNDHP becomes easier when the connectivity k increases.

6 Branch-and-cut algorithm for the *k*NDHP with $L = 4$ and $k = 2$

In this section, we present a branch-and-cut algorithm for the kNDHP when $L = 4$ and $k = 2$, based on the formulation presented in [24]. The formulation uses inequalities (1)–(6). First, we present the general framework of the algorithm and then present the separation procedures we have devised for the inequalities involved in the algorithm.

6.1 The general framework

The general framework of the algorithm is similar to the one presented before. To reinforce the linear relaxation of this problem, we add the rooted partition inequalities (25) in the cutting plane algorithm. The separation of the inequalities used in the branch-and-cut algorithm are performed in the following order

1. st-cut and st-node-cut inequalities,
2. rooted partition inequalities,
3. L-st-path-cut inequalities and L-st-node-path-cut inequalities (only for integral solutions).

We apply the rooted partition inequalities (25) for the rooted 2NDHP (that is, when the set of demands is rooted in a single node), and do not apply them when arbitrary demands are considered.

Notice that the L-st-path-cut and L-st-node-path-cut inequalities are separated only for integral solutions. Indeed, as we will see in the next subsection, these two families of inequalities can be efficiently separated when the solution x^* is integral.

In the following, we describe the separation algorithms we have devised for the inequalities (1)–(4) and the rooted partition inequalities (25).

6.2 Separation procedures

6.2.1 Separation of st-cut inequalities and st-node-cut

It is well-known that the separation of the st-cut inequalities (1) (resp. the st-node-cut inequalities (3)) reduces to computing a minimum weight cut in G (resp. in $G \setminus z$ for all $z \in V \setminus \{s, t\}$) with respect to weight vector $\overline{y}$. Indeed, there is a violated cut inequality (1) (resp. st-node-cut inequality (3)) if and only if the minimum weight of a cut, w.r.t. weight vector $\overline{y}$, is < 2 (resp. < 1). One can compute a minimum weight cut in polynomial time by using any minimum cut algorithm, and especially by using the Gomory-Hu algorithm [17] which computes the so-called Gomory-Hu cut tree. This algorithm consists in $|V| - 1$ maximum flow computations.

6.2.2 Separation of 4-st-path-cut and 4-st-node-path-cut inequalities

Now, we discuss the separation problem of 4-st-path-cut and 4-st-node-path-cut inequalities (2) and (4). As mentioned before, we consider the separation problem of these inequalities only in the case where the considered solution $\overline{x} \in \mathbb{R}^E$, is integral.

The idea is similar to the one presented in the proof of the formulation in [24]. Consider an edge subset $F \subseteq E$, and let G_F be the graph induced by F. First we compute a Dijkstra algorithm to obtain a shortest st-path (in number of hops), say P_0, in G. If $|P_0| > 4$, then we detect a violated 4-path-cut inequality. We define V_i, $i = 0, \ldots, 4$, as the subset of nodes at distance i from s in G, and $V_5 = V \setminus \left(\bigcup_{i=0}^{4} V_i \right)$. We add the corresponding 4-path-cut inequality induced by the partition $(V_1, \ldots, V_5)$ to the LP. If $|P_0| \leq 4$, then we look for a second shortest path in $G \setminus \{st\}$, say P_1, such that P_0 and P_1 are node-disjoint. If $|P_1| \leq 4$, then F induces a solution for the 2NDHP. If $|P_1| > 4$, there are two cases. The first case is when $|P_0| = 1$, that is $P_0 = (st)$, we define a 4-path-cut inequality in the same way as in the previous case, and we add the violated inequality to the LP. The second case is when $|P_0| > 1$, in that case we remove the nodes of P_0, say $v_i^{P_0}$, $i = 1, \ldots, |P_0|$, one by one, then we define the corresponding 4-node-path-cuts in $G \setminus v_i^{P_0}$ in the same way, and add them to the LP.

6.2.3 Separation of rooted partition inequalities

To separate inequalities (25), we use the separation heuristic presented in [25]. This heuristic has been implemented to run in polynomial time.

6.3 Computational results

The same computational environment presented in the previous section is used for these experiments. Note that the rooted partition inequalities (25) are only used for the instances with rooted demands.

The computational results are given in Tables 9 and 10. The entries of these two tables are the same as those of Section 5.3, except for Table 10, for which we add the entry

RP : the number of generated rooted partition inequalities.

We can see that for the rooted demands (Table 10), the algorithm has solved to optimality 7 instances out of 19 within the time limit. We can observe that the gaps obtained are quite large for most of the instances, but it is less than 30% for the relatively small instances. Table 10 also shows that a very large number of st-cut and 3-st-path-cut inequalities have been generated during the resolution, and a large number of st-node-cut and 3-st-node-path-cut inequalities have been generated for all the instances. We can also see that several rooted partition inequalities have been generated for some instances.

For the arbitrary demands, the algorithm has solved to optimality 6 instances, with graphs having up to 14 nodes and with 7 demands, and has spent all the CPU for the other instances. We can also see, as for the rooted demands, that a very large number of st-cut and 3-st-path-cut inequalities have been generated, and as much st-node-cut and 3-st-node-path-cut inequalities have been generated. We also note that the number of nodes in the branch-and-cut tree is quite large for the two types of demands. Also, the different gaps achieved are important for the big instances. Finally, we notice that for the arbitrary demands, the algorithm ran out of memory for 4 instances, and did not find a feasible solution.

Table 9 Results for $k = 2$, $L = 4$ and arbitrary demands

$\|V\|$	$\|D\|$	C-NC	LPC-NLPC	COpt	Gap	NSub	CPU
a 5	2	1	0	2314	0	1	0:00:01
a 10	3	17	15	2358	1.64	23	0:00:01
a 10	4	30	175	2773	10.1	186	0:00:01
a 10	5	15	326	3219	11.14	615	0:00:01
a 14	5	46	102	3326	7.18	356	0:00:01
a 14	7	195	1604	3796	9.72	6439	0:00:08
a 17	8	3589	45507	3079	31.71	608368	5:00:00
a 21	10	2361	37423	4720	40.59	334898	5:00:00
a 21	11	2893	36324	4770	41.44	334084	5:00:00
a 48	10	26795	19456	57349	87.33	154736	5:00:00
a 48	15	22343	32540	–	–	145813	5:00:00
a 48	24	17236	30632	–	–	245307	5:00:00
a 52	10	33319	16521	19769	74.6	107555	5:00:00
a 52	15	29114	15508	32592	81.2	124780	5:00:00
a 52	20	14288	33394	–	–	162161	5:00:00
a 52	26	10778	33464	–	–	198811	5:00:00

Table 10 Results for $k = 2$, $L = 4$ and rooted demands

$\|V\|$	$\|D\|$	C-NC	LPC-NLPC	RP	COpt	Gap	NSub	CPU
r 10	5	41	24	36	2358	1.35	17	0:00:01
r 10	7	18	18	22	2848	4.09	23	0:00:01
r 10	9	15	565	19	3481	12.28	588	0:00:01
r 14	5	520	444	0	2601	7.9	383	0:00:01
r 14	7	862	4177	882	2998	17.61	5820	0:00:36
r 14	10	622	3481	668	3662	7.04	10540	0:00:31
r 17	16	3202	83928	0	2700	22.65	380463	5:00:00
r 21	7	717	1272	0	1789	6.59	425	0:00:02
r 21	10	15721	35009	0	2570	25.91	431618	5:00:00
r 30	29	7391	200914	0	13549	54.59	163377	5:00:00
r 48	10	60181	46849	0	27557	77.47	81057	5:00:00
r 48	15	69824	71232	0	37962	82.38	77855	5:00:00
r 48	20	89805	86653	0	27814	72.68	70272	5:00:00
r 48	30	87560	158432	0	38629	79.03	105848	5:00:00
r 52	10	85786	59630	0	23916	83.1	62374	5:00:00
r 52	20	116638	109693	0	19426	72.81	50496	5:00:00
r 52	30	111091	174649	0	22143	72.07	63565	5:00:00
r 52	40	43333	262234	0	22378	70.09	109805	5:00:00
r 52	50	18217	281265	0	20731	64.4	120560	5:00:00

7 Conclusion

In this paper, we have studied the k-node-disjoint hop-constrained network design problem (kNDHP) when $L \in \{2, 3, 4\}$. We have introduced an integer programming formulation for the problem when $L \in \{2, 3\}$ and investigated the associated polytope. We have presented several classes of valid inequalities and presented conditions under which these inequalities define facets. Then, we have devised a branch-and-cut algorithm for solving the problem based on the inequalities we have presented before. In particular, we have discussed the separation problem of the st-cut, 3-st-path-cut, st-node-cut, 3-st-node-path-cut inequalities, as well as that of the Steiner SP-partition, Steiner partition, and double cut inequalities. Finally, we have presented branch-and-cut and computational results for the problem when $L = 3$ and $k = 3, 4, 5$ on one hand, and when $L = 4$ and $k = 2$ on the other hand.

The experiments we have done in this paper have shown that the branch-and-cut algorithm is quite efficient for solving the kNDHP when $L = 3$ and $k = 3, 4, 5$, and this, for both rooted and arbitrary sets of demands. They also pointed out that the large size instances are still difficult to solve within 5 hours of CPU time, but the gaps achieved, are in most cases quite interesting. Moreover, the experiments have shown the importance of Steiner SP-partition and double cut inequalities (17) and (11) are important in solving the problem, and that Steiner partition inequalities (19) seems to be less effective.

It should also be noticed that, contrarily to the survivable network design problem without hop constraints (or the kNCSP with $L \geq |V| - 1$), our experiments cannot permit to conclude on the impact of an increasing of the connectivity k on the resolution of the problem. In fact, previous experiments done for the survivable network design problem (see [3] and [29], for example) have concluded that the problem without considering hop constraints seems to become easier when k increases. In our case (the kNDHP with $L < |V| - 1$), the impact of the connectivity on the resolution is less clear. It even seems, when comparing the results for $L = 3$ and $L = 4$, that the kNDHP becomes more difficult to solve when L increases.

The computational study pointed out that a very large number of st-cut and 3-st-path-cut inequalities are generated during the resolution of the problem. This can be an issue since it yields the branch-and-cut algorithm to manage a huge pool of constraints and can imply an excessive CPU time consumption for constraints management. This can even yield the branch-and-cut algorithm to solve linear programs with a large number, but still polynomial, number of constraints. Finally, all this may prevent the algorithm from a good exploration of the branch-and-cut tree.

The above observations suggests that an efficient algorithm for the kNDHP requires a tighter formulation for the problem, which may efficiently include simultaneously both the disjoint paths and the hop contraints. Also, it may require a deeper investigation of the polytope of the problem in order to provide more facet defining inequalities and yield an efficient branch-and-cut algorithm.

For theoretical purposes, it should be interesting to study the polyope of the kNDHP in some special cases, like for example when the graph is series-parallel. Also, one could investigate the problem with respect to the distribution of the demands, since it may influence the polyhedral description of the solutions of the problem, and probably the efficiency of resolution algorithms.

Another question which would be of interest is to see whether one can use directed models for the kNDHP. This may provide stronger integer linear programming formulations. This is one of our research lines in the future.

Acknowledgements We would like to thank the anonymous referees for their valuable comments that permitted to correct some flaw in the previous version and improve the presentation of the paper.

References

1. Barahona F, Mahjoub AR (1995) On two-connected subgraph polytopes. Discret Math 147:19–34
2. Bendali F, Diarrassouba I, Mahjoub AR, Mailfert J (2010) The k edge-disjoint 3-hop-constrained paths polytope. Discret Optim 7:222–233
3. Bendali F, Diarrassouba I, Didi Biha M, Mahjoub AR, Mailfert J (2010) A branch-and-cut algorithm for the k-edge-connected subgraph problem. Networks 55:13–32
4. Botton Q, Fortz B, Gouveia L (2015) On the hop-constrained survivable network design problem with reliable edges. Comput Oper Res 64:159–167
5. Botton Q, Fortz B, Gouveia L, Poss M (2013) Benders decomposition for the hop-constrained survivable network design problem. INFORMS J Comput 25:13–26
6. Chimani M, Kandyba M, Ljubic I, Mutzel P (2010) Orientation-based models for 0, 1, 2-survivable network design: theory and practice. Math Program 124(1-2):413–439
7. Dahl G (1999) Notes on polyhedra associated with hop-constrained paths. Oper Res Lett 25:97–100
8. Dahl G, Foldnes N, Gouveia L (2004) A note on hop-constrained walk polytopes. Oper Res Lett 32:345–349
9. Dahl G, Gouveia L (2004) On the directed hop-constrained shortest path problem. Oper Res Lett 32:15–22
10. IBM, IBM ILOG CPLEX Optimization studio 12.5 documentation (2013). Available at: http://www-01.ibm.com/support/knowledgecenter/SSSA5p_12.5.1/maps/ic-homepage.html
11. Diarrassouba I (2009) Survivable network design problems with high connectivity requirements, PhD Thesis, Université Blaise Pascal, France
12. Diarrassouba I, Kutucu H, Mahjoub AR (2016) Two node-disjoint hop-constrained survivable network design and polyhedra. Networks 67:316–337
13. Diarrassouba I, Gabrel V, Mahjoub AR, Gouveia L, Pesneau P (2016) Integer programming formulations for the k-edge-connected 3-hop-constrained network design problem. Networks 67:148–169
14. Didi Biha M, Mahjoub AR (2004) The k-edge connected subgraph problem I: polytopes and critical extreme points. Linear Algebra Appl 381:117–139
15. Didi Biha M, Mahjoub AR (1996) K-edge connected polyhedra on series-parallel graphs. Oper Res Lett 19:71–78
16. Edmonds J, Karp RM (1972) Theoretical improvements in algorithmic efficiency for network flow problems. J ACM 19:248–264
17. Gomory RE, Hu TC (1961) Multi-terminal network flows. JSoc Ind Appl Math 9:551–570
18. Gouveia Luis, Leitner Markus (2017) Design of survivable networks with vulnerability constraints. Eur J Oper Res 258(1):89–103
19. Gouveia L, Patricio P, de Sousa A (2005) Compact models for hop-constrained node survivable network design, an application to MPLS, telecommunications planning: innovations in pricing, network design and management. Springer 33:167–180
20. Grötschel M, Monma CL, Stoer M (1991) Polyhedral approaches to network survivability. Series in Discrete Mathematics & Theoretical Computer Science 5:121–141
21. Grötschel M, Monma CL (1990) Integer polyhedra arising from certain network design problems with connectivity constraints. SIAM J Discret Math 3:502–523
22. Grötschel M, Monma CL, Stoer M (1992) Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. Oper Res 40:309–330
23. Grötschel M, Monma CL, Stoer M (1995) Polyhedral and computational investigations for designing communication networks with high survivability requirements. Oper Res 43:1012–1024
24. Huygens D, Mahjoub AR (2007) Integer programming formulations for the two 4-hop-constrained paths problem. Networks 49:135–144
25. Huygens D, Labbé M., Mahjoub AR, Pesneau P (2007) The two-edge connected hop-constrained network design problem: valid inequalities and branch-and-cut. Networks 49:116–133
26. Huygens D, Mahjoub AR, Pesneau P (2004) Two edge-disjoint hop-constrained paths and polyhedra. SIAM J Disc Math 18:287–312
27. Kerivin H, Mahjoub AR (2015) Design of survivable networks: a survey. Networks 46:1–21
28. Kerivin H, Mahjoub AR, Nocq C (2004) (1,2)-survivable networks: facets and branch and cut, The sharpest cut. In: Grötschel M (ed) MPS/SIAM optimization, pp 121–152
29. Mahjoub M, Diarrassouba I, Mahjoub AR, Taktak R (2017) The survivable k-node-connected network design problem: valid inequalities and Branch-and-Cut. Comput Ind Eng 112:690–705
30. Mahjoub AR (1994) Two-edge connected spanning subgraphs and polyhedra. Math Program 64:199–208
31. Mahjoub AR, Nocq C (1999) On the linear relaxation of the 2-node connected subgraph polytope. Discret Appl Math 95(1–3):389–416
32. Mahjoub AR, Simonetti L, Uchoa E (2011) Hop-level flow formulation for the hop constrained survivable network design problem. Lect Notes Comput Sci 6701:176–181
33. Menger K (1927) Zur allgemeinen kurventhorie. Fundamanta Mathematicae 10:96–115
34. TSPLIB (1991). Available at: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/