INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 32 (2025) 918–960
DOI: 10.1111/itor.13511

Design of survivable networks with low connectivity requirements

Fatmah Almathkour[a,*], Youcef Magnouche[b], A. Ridha Mahjoub[a,c] and Raouia Taktak[d,e]

[a] *Department of Statistics and Operations Research, College of Science, Kuwait University, Safat 13060, Kuwait*
[b] *Huawei Technologies, France Research Center, 18 Quai du Point du Jour, Boulogne-Billancourt, Paris, France*
[c] *LAMSADE, UMR CNRS 7243, Université Paris-Dauphine PSL, Paris, France*
[d] *ISIMS, Université de Sfax, Sfax, Tunisia*
[e] *Sm@rts, Centre de Recherche en Numérique de Sfax (CRNS), Sfax, Tunisia*
E-mail: fatmah.almazkoor@ku.edu.kw [Almathkour]; youcef.magnouche@huawei.com [Magnouche]; ridha.mahjoub@ku.edu.kw [Mahjoub]; raouia.taktak@isims.usf.tn [Taktak]

Received 4 December 2023; received in revised form 2 May 2024; accepted 21 June 2024

Abstract

In this paper, we consider the survivable network design problem when the connectivity types are in {0, 1, 2, 3}. The problem has wide applications in telecommunication networks. We consider the problem from a polyhedral point of view. We describe several classes of valid inequalities and give necessary conditions and sufficient conditions for these inequalities to be facet defining. We also develop separation routines for these inequalities. Using these results, we devise a branch-and-cut algorithm along with an extensive computational study is presented.

Keywords: survivable networks; low-connectivity; integer programming; valid inequality; facet; polytope; branch-and-cut

1. Introduction

With the introduction of optical fiber technology, survivability became a main issue in the design of telecommunication networks. Nowadays telecommunication networks are sparse in many real situations. Moreover, they have almost unlimited capacity and are transporting huge quantities of data. As a consequence, in the event of a failure on one (or more) of its links or centers, an important amount of information may be lost and this would be a disaster for the operator. For this, now telecommunication networks have to be sufficiently survivable in the sense that the network must remain functional in the event of a failure (see Larsson, 2014).

* Corresponding author.

Survivability is generally expressed in terms of connectivity in the network. It is required that between each pair of nodes there is a certain number of edge (or node)-disjoint paths so that if some links (or centers) fail, the traffic continues to be routed through the nonaffected paths. The first general survivability model has been introduced by Steiglitz et al. (1969). In what follows, we present a second more practical model given by Grötschel et al. (1995).

1.1. The survivable network design problem

Let $G = (V, E)$ be an undirected graph. Suppose that with each node $v \in V$ it is associated an integer $r(v) \geq 0$ called the *connectivity type* of v. A subgraph H of G is said to be *survivable*, if between every pair of nodes s, t of H, there exist at least

$$r(s, t) = \min\{r(s), r(t)\}$$

edge (or node)-disjoint paths. The connectivity type $r(v)$ can be seen as the minimum number of links that must connect v to the network. If the connectivity requirements are given in terms of edges (nodes) then we talk about *edge (node)-survivability*.

If the edges of G have weights, then the *survivable network design problem (SNDP)* is to find a survivable subgraph with minimum weight. Different variants of this problem have been considered in the literature (Stoer, 1992; Grötschel et al., 1995; Kerivin and Mahjoub, 2005a). Most of these variants concern the uniform case, that is when the connectivity types are the same and equal to some positive integer $k \geq 0$ for all the nodes. In that case, the problem is nothing but the k-edge (resp. k-node) connected subgraph problem. The case when $k = 2$ has seen particular attention for both the edge and node variants. Here the network must verify the 2-connectivity conditions, that is to say, between every pair of nodes there are at least two edge (node)-disjoint paths. This topology has shown very efficient in sufficiently securing the network. Indeed, if one of the paths (links, centers) fails, the traffic can be rerouted on the second one. However, in some situations, the nodes (centers) of the telecommunication network may not have the same importance, and some of them may need a higher degree of survivability. For instance, it may happen that two links fail at the same time. If these links belong to two disjoint paths between two nodes s and t, in order to remain functional, the network must contain a third path, disjoint of the two first ones, on which the traffic can be rerouted between s and t. However, in practice, the highest survivability degree that might be required is three. This corresponds to some specific networks, like military telecommunication networks.

1.2. Our contributions

In this paper, we consider this variant of the survivable network design problem. More precisely, we will consider the edge case where $r(v) \in \{0, 1, 2, 3\}$ for all $v \in V$. The problem here is then the *edge survivable network design problem (ESNDP)* with $r(v) \in \{0, 1, 2, 3\}$. To the best of our knowledge, this more general variant has not been considered in the literature. We study the problem from a polyhedral point of view. We describe various classes of valid inequalities and give necessary conditions and sufficient conditions for these inequalities to be facet defining. We also devise separation routines for these inequalities. Using this, we propose a branch-and-cut algorithm along with an extensive computational study.

1.3. Literature review and related works

The ESNDP is NP-hard in general since some special cases of it like the 2-edge connected subgraph problem ($r(v) = 2$ for all $v \in V$) and the Steiner tree problem ($r(v) \in \{0, 1\}$ for all $v \in V$) are NP-hard (see Garey and Johnson1979). It has been shown polynomially solvable in some special cases. Takamizawa et al. (1982), show that if $r \in \{0, 1\}$ the ESNDP can be solved in polynomial time in series–parallel graphs. Also, Winter (1986, 1985, 1987a, 1987b) gives polynomial time algorithms for solving the ESNDP in series–parallel and Halin graphs when $r \in \{0, 2\}^V$. Baïou and Mahjoub (1997) and Mahjoub and Pesneau 2008 give in this case a complete description of the associated polytope when the underlying graph is series–parallel and a Halin graph, respectively. Biha and Mahjoub 2000 extended Baïou and Mahjoub's result to the case when $r \in \{0, k\}^V$. Winter (1985, 1987a) also gives a polynomial time algorithm for the ESNDP in Halin graphs when $r \in \{0, 3\}^V$. Monma and Shallcross (1989) propose a local search heuristic for the problem when $r \in \{1, 2\}^V$.

Kerivin and Mahjoub (2005b) show that when $r(v)$ is even for all $v \in V$, and the graph is series–parallel, the associated polytope is given by the so-called cut inequalities and trivial inequalities. Biha and Mahjoub (1996) show a similar result when $r(v) = k$ for all $v \in V$ for some integer $k \geq 1$. They give in both, edge and node cases, a complete description of the related polytope in series–parallel graphs. Grötschel et al. (1992a, 1992b) consider the ESNDP with $r(v) \in \{0, 1, 2\}$. They characterize when the basic inequalities (the cut and trivial inequalities) are facet defining. Moreover, they describe further classes of valid inequalities and give necessary conditions and sufficient conditions for these inequalities to define facets. Grötschel et al. (1992b) propose a cutting plane algorithm for the problem in this case.

Kerivin et al. (2004) study the variant of the ESNDP for $r(v) \in \{1, 2\}$. This case is of particular interest to telecommunication operators. They describe valid inequalities along with separation algorithms and propose a branch-and-cut algorithm. The same variant is also considered in Biha et al. (2008). The authors consider the linear relaxation of the associated polytope given by the so-called cut and partition inequalities together with the trivial inequalities. They give sufficient conditions for this polytope to be an integer when the underlying graph is series–parallel. Moreover, thanks to a polynomial time separation algorithm for the partition inequalities given in Kerivin and Mahjoub (2002), they obtain a polynomial time algorithm for the ESNDP in this case in a subclass of series–parallel graphs. They also introduce a new class of facets in this case. Slama (2008) investigates the case when $r(v) \in \{1, k\}$, where k is a fixed integer ≥ 2. She discusses the polytope given by the so-called SP-partition inequalities together with the trivial inequalities. Some structural properties of this polytope are given. As a consequence, she shows that the polytope is integral in a subclass of series–parallel graphs containing the outerplanar graphs. Moreover, some valid inequalities for $r \in \{1, k\}^V$ are introduced. These results generalize those given by Biha et al. (2008) for the case $r \in \{1, 2\}^V$.

Raghavan (2004) considers the ESNDP when $r \in \{0, 1, 2\}^V$. Linear time algorithms are proposed for both the edge and node versions when the underlying graph is series–parallel. A dual-ascent algorithm is proposed for the problem by Magnanti and Raghavan (1999).

Goemans and Bertsimas (1993) study the ESNDP when each edge can be used more than once. They introduce what is called the parsimonious property defined by the solutions of the linear relaxation of the problem. They also propose two tree-based heuristics. Using these, together with the parsimonious property, they show that for $r \in \{0, 1, 2\}^V$, $r \in \{0, 1, 3\}^V$ and $r \in \{0, k\}^V$ for integer $k \geq 1$, the optimal value given by either of the heuristics is within twice of the optimal value of the problem.

The SNDP also has applications in network slicing. The 5G mobiles are based on the concept of network slicing. This new network architecture permits the partitioning of the network into subnetworks (network slices) and the multiplexing of these subnetworks on the same network infrastructure. Each part of the network has its specific requirements in terms of survivability, routing, and security (Gajić et al., 2023; Rezk et al., 2023).

The closely related node survivable network design problem (NSNDP) has also been extensively investigated, in particular in the uniform case, that is to say, when $r(v) = k$ for all $v \in V$ for some integer $k \geq 1$ (see Grötschel et al., 1992a, 1992b, 1995; Mahjoub and Nocq, 1999; Kerivin and Mahjoub, 2005a; Diarrassouba et al., 2016, 2017; Mahjoub, 2017).

A huge amount of work has been conducted to devise approximation algorithms for the NSNDP (Hochbaum, 1997, Ravi and Williamson1997). Fleischer (2001) devises a 2-approximation algorithm for NSNDP when $r \in \{0, 1, 2\}^V$. Fleischer's algorithm (see also Fleischer et al., 2006) extends the 2-approximation given by Jain (2001) for the general survivable network design problem. It also improves on a primal-dual 3-approximation algorithm for the NSNDP with $r \in \{0, 1, 2\}^V$.

Recently, Chen et al. (2022) consider a more general problem called the group edge connected SNDP where survivability is required between groups instead of point to point as in the classical case. The authors generalize some approximation results. In particular, they give a randomized polynomial approximation algorithm for the problem.

Gouveia and Leitner (2017) discuss the survivable network design problem, which consists of finding a minimum weight subgraph containing, between each pair of nodes, two bounded paths. The two bounds may be different, and the second path must be ensured after at most $k - 1$ link failures. Here k is an integer ≥ 1. This problem generalizes the ESNDP. It also provides a more appropriate quality of service (QoS) for routing demands. In fact, limiting the length of the routing paths may reduce significantly the routing cost in the network. QoS also becomes an important issue in the design of telecommunication networks (see Wierzbicki and Burakowski, 2011). A similar preplanned protection (survivability) approach is discussed by Babarczi and Tapolcai (2012). Applications of the ESNDP can also be found in multiagent systems and unmanned aerial vehicles (UAVs) swarms. Leu and Tang (2019) propose an approach to survivable networks in which they bring swarm intelligence. They devise a genetic algorithm to optimize the movements of a UAV swarm towards maintaining communication between the ground agents. Chen et al. (2022) consider the problem of constructing survivable networks for consensus in multiagent systems. They discuss, in particular, the case when the survivable network is a k-edge connected rooted digraph and k-node connected rooted digraph (see also Duhamel and Santos, 2024; Patricio, 2024).

Fouilhoux et al. (2012) discuss survivability in hierarchical telecommunication networks. They consider the problem of locating concentrators and linking concentrators in a 2-edge connected backbone network. They devise a branch-and-cut algorithm for the problem. Blanco and Puerto (2022) study the closely related hub location problem. They also devised a branch-and-cut algorithm for the problem. Further related works can be found in Olivera et al. (2010), Eshoul and Mouftah (2012), Wang (2012), and Ameln et al. (2021).

1.4. Organization of the paper

The paper is organized as follows. In the following section, we give a formulation for the ESNDP and introduce the associated polytope. In Section 3, we describe some valid inequalities and give necessary conditions and sufficient conditions for these inequalities to be facet defining. These inequalities are generalizations of known ones. We also introduce a new class of valid inequalities. In Section 5, we present the branch-and-cut algorithm, the separation routines for the different classes of inequalities as well as some reduction operations. We discuss our experimental results in Section 6, and Section 7 is devoted to some concluding remarks.

1.5. Notations

In the rest of this section, we give more notations and definitions. We consider loopless, finite, and undirected graphs. We denote a graph by $G = (V, E)$, where V is the node set and E is the edge set. If e is an edge between two nodes u and v, then we write $e = uv$ or $e = \{u, v\}$, and $G\backslash e$ will denote the subgraph obtained by deleting edge e. For $W \subset V$, the set of edges having one node in W and the other in $V\backslash W$ is called a *cut* and denoted by $\delta(W)$. If $W = \{v\}$ for some $v \in V$, then we write $\delta(v)$ for $\delta(\{v\})$. A *partition* of V is a collection of pairwise disjoint subsets of V, $V_1, \dots, V_p$ whose union is V. If $V_1, \dots, V_p$ is a partition of V, then we denote by $\delta(V_1, \dots, V_p)$ the set of edges having their end nodes in different sets of the partition. If U and W are two subsets of V, then we write $[U, W]$ for the set of edges between U and W. If $U \subseteq V$ then $E(U)$ denotes the set of edges having both nodes in U. And for $F \subseteq E$, $V(F)$ denotes the set of nodes of the edges of F.

If $U \subseteq V$, then $G(U)$ denotes the subgraph of G induced by U, that is the subgraph $(U, E(U))$. For $F \subseteq E$, the $0-1$ vector x^F such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ if not is called the *incidence vector* of F. For a vector $x \in \mathbb{R}^E$ and $F \subseteq E$, we let $x(F) = \sum_{e\in F} x(e)$. If $\pi = (V_1, \dots, V_p)$ is a partition of V, then we also write $\delta(\pi)$ for $\delta(V_1, \dots, V_p)$.

A graph is called *series–parallel* if it can be obtained from an edge by means of the application of the following operations:

θ_1: add a parallel edge (duplicate an edge),

θ_2: add a node of degree 2 (subdivide an edge).

It has been shown that a graph is series–parallel if and only if it is not contractible to K_4 (the complete graph on four nodes). Recall that a graph G is said to be contractible to a graph H if H can be obtained from G by contraction and deletion of edges.

A graph is *outerplanar* if it is planar and if all the nodes can be disposed of on the exterior cycle. Remark that an outerplanar graph is series–parallel.

A *Halin graph* is a graph that consists of a cycle and a tree without a node of degree 2 whose leaves are precisely the nodes of the cycle.

2. Formulation and associated polytope

Let $G = (V, E)$ be a graph with connectivity types $r(v) \in \{0, 1, 2, 3\}$ for all $v \in V$. If $W \subset V$, let $r(W) = \max\{r(v), v \in W\}$ and $con(W) = \min\{r(W), r(V\backslash W)\}$. So $con(W)$ can be interpreted as the connectivity type of W, that is the minimum number of links that must connect W to $V\backslash W$.

Let $c(e)$, $e \in E$ be weights associated with the edges of E. By Menger theorem (Menger, 1927), the ESNDP is equivalent to the following integer program.

$$\min \sum_{e \in E} c(e)x(e) \tag{1}$$

$$x(\delta(W)) \geq con(W) \qquad \text{for all } W \subset V, W \neq \emptyset, \tag{2}$$

$$0 \leq x(e) \leq 1 \qquad \text{for all } e \in E, \tag{3}$$

$$x(e) \in \{0, 1\} \qquad \text{for all } e \in E. \tag{4}$$

Inequalities (2), called *cut inequalities*, ensure the connectivity requirements between the pairs of nodes. Inequalities (3) are the trivial inequalities, and constraints (4) are the integrality constraints.

Let ESNDP(G) be the polytope the extreme points of which are the solutions of (2)–(4). ESNDP(G) will be called the *edge survivable network design polytope* of G. An edge e is said to be *essential* if e must belong to every solution of the ESNDP. Let E^* be the set of essential edges. We have the following result characterizing the dimension of ESNDP(G), which is easily seen to be true.

Theorem 1. *The dimension of* ESNDP(G) *is* $|E| - |E^*|$.

As a consequence of Theorem 1, ESNDP(G) is full dimensional if and only if G has no essential edges.

Note that the essential edges can be detected in polynomial time. For each edge $e = uv$, one can consider the max-flow problem from u to v with the capacities $c(e) = 0$ and $c(e') = 1$ for all $e' \in E \setminus \{e\}$. If the max-flow is of value equal to $\min(r(u) - r(v)) - 1$, this means that the minimum cut separating u and v contains exactly $\min(r(u) - r(v))$ edges, and consequently all the edges of the cut (including $e = uv$) are essential. If e is essential, such a cut should appear.

3. Valid inequalities and facets

In this section, we describe some valid inequalities for ESNDP(G) and give necessary conditions and sufficient conditions for these inequalities to define facets.

Given a graph $G = (V, E)$ and a partition $\pi = (V_1, \ldots, V_p)$ of V, we let G_π denote the graph obtained from G by contracting $V_1, \ldots, V_p$, that is to say, the graph obtained from G by replacing the sets V_i, $i = 1, \ldots, p$ by nodes. Note that G_π has p nodes and may contain parallel edges.

Let $G = (V, E)$ be a graph and $r \in \{0, 1, 2, 3\}^V$ a system of connectivity types associated with the nodes of V. If $\pi = (V_1, \ldots, V_p)$ is a partition of V, then we let $p_i(\pi)$ stand for the number of sets V_j in π with $con(V_j) = i$ for $i = 0, 1, 2, 3$. That is $p_i(\pi) = |\{j \mid con(V_j) = i\}|$. Also, for a partition $\pi = (V_1, \ldots, V_p)$, we let $r(\pi) = \max\{con(V_i), i = 1, \ldots, p\}$.

3.1. SP-partition inequalities

Chopra (1994) introduces a class of valid inequalities for the k-edge connected subgraph problem when the underlying graph is outerplanar, k is odd, and each edge can be used more than once. He shows that, given an outerplanar graph $G = (V, E)$ and an odd integer $k \geq 1$, if $\pi = (V_1, \ldots, V_p)$ is a partition of V, then the inequality

$$x(\delta(\pi)) \geq \left\lceil \frac{k}{2} \right\rceil p - 1 \tag{5}$$

is valid for the associated polytope.

Biha and Mahjoub (1996) extend inequalities of type (5) for series–parallel graphs and when each edge can be used at most once and call them *SP-partition inequalities* (SP stands for series-parallel).

Inequalities (5) may be nonvalid for non-series–parallel graphs. In fact, consider graph K_4, the complete graph on four nodes, and suppose each node is of connectivity type 3. Clearly, the graph itself is a solution to the 3-edge-connected subgraph problem. By considering the trivial partition, where each node is a set of the partition, it follows that the right-hand side of (5), in this case, is 7 (here $k = 3$ and $p = 4$) whereas the graph only contains six edges.

Kerivin (2000) generalizes inequalities (5) for the more general survival network design problem. He shows that, given a series–parallel graph $G = (V, E)$ and a system of connectivity types $r \in N^V$, for a partition $\pi = (V_1, \ldots, V_p)$ of V such that $r(V_i) \geq 1$ for $i = 1, \ldots, p$, the inequality

$$x(\delta(\pi)) \geq \begin{cases} \sum\limits_{i=1}^{r(\pi)} \lceil \frac{i}{2} \rceil p_i(\pi) - 1, & \text{if } r(\pi) \text{ is odd,} \\ \sum\limits_{i=1}^{r(\pi)} \lceil \frac{i}{2} \rceil p_i(\pi), & \text{if } r(\pi) \text{ is even,} \end{cases} \tag{6}$$

is valid for the survivable network design polytope.

In what follows we give the SP-partition inequalities for the ESNDP(G) and describe conditions for them to be facet defining.

Let $G = (V, E)$ be a graph. Let $\pi = (V_1, \ldots, V_p)$ be a partition of V such that $r(V_i) \geq 1$ for $i = 1, \ldots, p$ and G_π is series–parallel. From (6), it follows that, when $r \in \{0, 1, 2, 3\}^V$, the inequality

$$x(\delta(\pi)) \geq \begin{cases} p + p_3(\pi) - 1, & \text{if } r(\pi) = 3, \\ p, & \text{if } r(\pi) = 2, \\ p - 1, & \text{if } r(\pi) = 1. \end{cases} \tag{7}$$

is valid for the ESNDP(G).

In fact, as $r \in \{0, 1, 2, 3\}^V$ and $r(V_i) \geq 1$ for $i = 1, \ldots, p$, we have that $r(\pi) \in \{1, 2, 3\}$.

If $r(\pi) = 3$, then $r(\pi)$ is odd, and by (6), we obtain that $x(\delta(\pi)) \geq \sum_{i=1}^{3} \lceil \frac{i}{2} \rceil p_i(\pi) - 1 = p_1(\pi) + p_2(\pi) + 2p_3(\pi) - 1$. As $p_1(\pi) + p_2(\pi) + p_3(\pi) = p$, it follows that $x(\delta(\pi)) \geq p + p_3(\pi) - 1$.

If $r(\pi) = 2$, then $r(\pi)$ is even, and (6) yields $x(\delta(\pi)) \geq \sum_{i=1}^{2} \lceil \frac{i}{2} \rceil p_i(\pi) = p_1(\pi) + p_2(\pi)$. As here, $p = p_1(\pi) + p_2(\pi)$, the statement follows:

If $r(\pi) = 1$, then $r(\pi)$ is odd, and by (6) it follows that $x(\delta(\pi)) \geq p_1(\pi) - 1 = p - 1$.

Inequalities of type (7) will be also called *SP-partition inequalities*.

Given a graph G, an *articulation node* of G (also called *cut vertex*) is a node whose removal increases the number of connected components (see Fig. 1). If $G = (V, E)$ has an articulation node

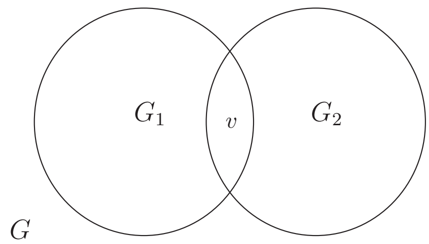

Fig. 1. A graph decomposing into two pieces by one articulation node.

v, then G decomposes into two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $V_1 \cup V_2 = V$, $E_1 \cup E_2 = E$, $V_1 \cap V_2 = \{v\}$, and $E_1 \cap E_2 = \emptyset$. The subgraphs G_1 and G_2 are called the *pieces* of G.

The following lemma will be useful in the sequel.

Lemma 1. *Let $\pi = (V_1, \dots, V_p)$ be a partition of V such that G_π is connected but not 2-node connected. Suppose that V_1 corresponds to the articulation node of G_π. Suppose that $V_1, \dots, V_q$ and $V_{q+1} \dots, V_p$ for some $2 \le q \le p$ are the node sets of the two pieces of G_π. Let $\pi_1 = (V_1 \cup \bigcup_{i=2}^{q} V_i, V_{q+1}, \dots, V_p)$ and $\pi_2 = (V_1 \cup \bigcup_{i=q+1}^{p} V_i, V_2, \dots, V_q)$. Then $r(\pi) = \max(r(\pi_1), r(\pi_2))$.*

Proof. Suppose $r(\pi) = t$ for some $1 \le t \le 3$. Then there is $i_0 \in \{1, \dots, p\}$ such that $r(V_{i_0}) = t$ and $r(V \setminus V_{i_0}) \ge t$. Suppose without loss of generality that $i_0 \in \{1, q+1, \dots, p\}$. If $r(V \setminus V_{i_0}) = t$, then $r(V_j) \le t$ for all $j \in \{1, \dots, p\} \setminus \{i_0\}$ and $r(V_{j_0}) = t$ for some $j_0 \in \{1, \dots, p\} \setminus \{i_0\}$. If $j_0 \in \{1, q+1, \dots, p\} \setminus \{i_0\}$ (resp. $j_0 \in \{1, 2, \dots, q\}$), then $r(\pi_2) = t$ (resp. $r(\pi_1) \le t$) and the statement follows. Now suppose that $r(V \setminus V_{i_0}) \ge t+1$. Then there exists a unique $k_0 \in \{1, \dots, p\} \setminus \{i_0\}$ such that $r(V_{k_0}) \ge t+1$. Otherwise, we would have $r(\pi) \ge t+1$, which contradicts the hypothesis. Hence, the connectivity requirements between the sets V_i do not change if we set $r(V_{k_0}) = t$ (instead of $t+1$). In this case, we will have $r(V \setminus V_{i_0}) = t$ and the previous case applies. □

Theorem 2. *Inequality (7) defines a facet of* ESNDP(G) *only if*

1. *$G(V_i)$ is connected for $i = 1, \dots, p$.*
2. *G_π is 2-node connected.*
3. *For every nonessential edge e, if $G_\pi \setminus e$ is not 2-node connected and $G_{\pi_1} = (W_1, E_1)$ and $G_{\pi_2} = (W_2, E_2)$ are the two pieces of $G_\pi \setminus e$ with $W_1 = \{w_1, \dots, w_q\}$, $W_2 = \{w_1, w_{q+1}, \dots, w_p\}$ (i.e., $W_1 \cap W_2 = \{w_1\}$) such that w_i corresponds to V_i, for $i = 1, \dots, p$, and $\pi_1 = (V_1 \cup \bigcup_{i=2}^{q} V_i, V_{q+1}, \dots, V_p)$ and $\pi_2 = (V_1 \cup \bigcup_{i=q+1}^{p} V_i, V_2, \dots, V_q)$, then either $r(\pi_1) = 1$ or $r(\pi_2) = 1$ or both.*
4. *Given a nonessential edge e_0 of $G(V_i)$ for some $i \in \{1, \dots, p\}$, then for every partition (V_i^1, V_i^2) of V_i with $e_0 \in [V_i^1, V_i^2]$, we have $[V_i^1, V_i^2] \setminus \{e_0\} \neq \emptyset$ (see Fig. A1).*

Proof. See Appendix A.1. □

In what follows, we give sufficient conditions for inequality (7) to be facet defining.

Theorem 3. *Suppose that $con(V_i) = 3$ for $i = 1, \dots, q$, $1 \le q \le p$ and $con(V_i) = 1$ for $i = q+1, \dots, p$. Then, inequality (7) defines a facet of ESNDP(G) if*

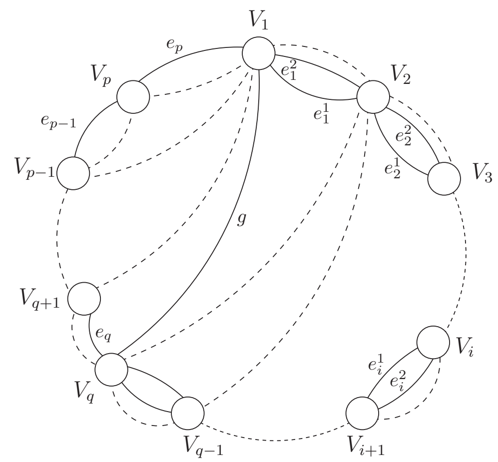

Fig. 2. Partition $\pi = (V_1, \ldots, V_q, V_{q+1}, \ldots, V_p)$.

1. *$G(V_i)$ is 4-edge connected for $i = 1, \ldots, q$ and 2-edge connected for $i = q + 1, \ldots, p$.*
2. $\left|[V_i, V_{i+1}]\right| \geq 2$*, for* $i = 1, \ldots, q - 1$,

   $\left|[V_i, V_{i+1}]\right| \geq 1$*, for* $i = q, \ldots, p$ *(mod p),*

   $\left|[V_1, V_q]\right| \geq 1$.
3. *G_π is series–parallel.*
4. *$G(\pi')$ (resp. $G(\pi'')$) is 3-edge connected (resp. 2-edge connected), where $\pi' = (V_1, \ldots, V_q)$ (resp. $\pi'' = (V_q, \ldots, V_p, V_1)$).*

Proof. First, by the conditions above, it can be seen that G has no essential edges, and hence ESNDP(G) is fully dimensional. Here, we have $p_3(\pi) = q$, and inequality (7) can be written as

$$x(\delta(V_1, \ldots, V_p)) \geq p + q - 1. \tag{8}$$

Denote by $ax \geq \alpha$ inequality (8), and let $bx \geq \beta$ be a facet-defining inequality of ESNDP(G) such that $\{x \in \text{ESNDP}(G) \mid ax = \alpha\} \subseteq \{x \in \text{ESNDP}(G) \mid bx = \beta\}$. It suffices to show that there is a scalar ρ such that $b = \rho a$. First, we will show that $b(e) = \rho$ for all $e \in [V_i, V_j]$, $i, j \leq q$, for some $\rho \in \mathbb{R}$.

By Condition (2), $[V_i, V_{i+1}]$ contains at least two edges (resp. one edge) for $i = 1, \ldots, q - 1$ (resp. $i = q, \ldots, p$) (mod p). Let e_i^1, e_i^2 be two edges of $[V_i, V_{i+1}]$ for $i = 1, \ldots, q - 1$ and e_i an edge of $[V_i, V_{i+1}]$ for $i = q, \ldots, p$ (mod p). Also by Condition (2), there exists an edge, say g, between V_1 and V_q (see Fig. 2). Remark that, as G is series–parallel, there is no edge between sets V_i and V_j where $2 \leq i \leq q - 1$ and $q + 1 \leq j \leq p$. Consider the edge sets

$$S_1 = \{e_i^1, e_i^2, i = 1, \ldots, q - 1\} \cup \{e_i, i = q, \ldots, p\} \cup \bigcup_{i=1}^{p} E(V_i),$$

$S_2 = (S_1 \setminus \{e_i^1\}) \cup \{g\}, i \in \{1, \dots, q-1\},$

$S_3 = (S_1 \setminus \{e_j\}) \cup \{g\}, j \in \{q, \dots, p\}.$

It is easily seen that S_1, S_2, and S_3 are all solutions of the ESNDP and $ax^{S_i} = \alpha$ for $i = 1, 2, 3$. Hence, $bx^{S_i} = \beta$ for $i = 1, 2, 3$, implying that $b(e_i^1) = b(e_j) = b(g)$. By symmetry, it follows that $b(e) = b(e') = b(g)$ for all $e, e' \in [V_i, V_{i+1}]$, for $i = 1, \dots, p$.

Therefore,

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

If f is an edge of $[V_i, V_j]$ with $i + 1 \leq j$ and $i, j \leq q$, then by considering the sets S_1 and $\left(S_1 \setminus \{e_i^1\}\right) \cup \{f\}$ (where S_1 is the set introduced above), we can show along the same line that

$$b(f) = b(e_i^1) = \rho.$$

In consequence, we obtain that

$$b(e) = \rho \begin{array}{l} \text{for all } e \in [V_i, V_j], i, j \leq q-1, \\ \text{for all } e \in [V_i, V_{i+1}], i = q, \dots, p \, (\text{mod}\, p). \end{array} \tag{9}$$

Now we will show that $b(e) = b(e')$ for all $e, e' \in [V_i, V_j]$, $q \leq i, j, \leq p$, $j \geq i + 1$. Let f be an edge between V_i and V_j, $i + 1 \leq j$, $q \leq i, j \leq p + 1$ (mod p). Let $S_4 = (S_1 \cup \{f\}) \setminus \{e_i\}$. We have that S_4 is a feasible solution, and $ax^{S_4} = \alpha$. Therefore, $bx^{S_4} = bx^{S_1} = \beta$, which implies that $b(f) = b(e_i)$. As by (9) $b(e_i) = \rho$, it follows that $b(f) = \rho$. Therefore,

$$b(e) = \rho \;\text{ for all }\; e \in \delta(\pi). \tag{10}$$

Recall that there are no edge sets V_i and V_j, $2 \leq i \leq q - 1$, $q + 1 \leq j \leq p$.

Now, let e be an edge of $G(V_i)$. Since $G(V_i)$ is 4-edge connected, the edge set $S_5 = S_1 \setminus \{e\}$ is still a feasible solution for the ESNDP. As $ax^{S_5} = \alpha$, and hence $bx^{S_1} = bx^{S_5} = \beta$, we have that $b(e) = 0$. So this, together with (9) and (10), yields

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

We then have $b = \rho a$, which ends the proof. □

3.2. *F*-partition inequalities

Mahjoub (1994) introduced a class of valid inequality for the 2-edge connected subgraph polytope. This can be presented as follows.

Let $\pi = (V_0, V_1, \dots, V_p)$ be a partition of V, and let $F \subseteq \delta(V_0)$ with $|F|$ odd. Suppose that $con(V_i) = 2$ for $i = 1, \dots, p$. The following inequalities are valid for the ESNDP(G):

$$\begin{aligned} x(\delta(V_i)) &\geq 2, && \text{for } i = 1, \dots, p, \\ -x(e) &\geq -1, && \text{for all } e \in F, \\ x(e) &\geq 0, && \text{for all } e \in \delta(V_0) \setminus F. \end{aligned}$$

By summing these inequalities, we obtain that

$$2x(\delta(\pi) \setminus F) \geq 2p - |F|.$$

By dividing by 2 and rounding up the right-hand side, we get

$$x(\delta(\pi) \setminus F) \geq p - \left\lfloor \frac{|F|}{2} \right\rfloor. \tag{11}$$

Inequalities of type (11) are called *F-partition inequalities*.

In what follows, we will discuss a generalization of these inequalities.

Suppose $r(v) \in \{0, 1, 2, 3\}$ for all $v \in V$ and consider again a partition $\pi = (V_0, V_1, \dots, V_p)$ of V such that $r(V_i) \geq 1$ for $i = 1, \dots, p$. Let F be an edge subset of $\delta(V_0)$ (not necessarily odd). The following inequalities are valid for ESNDP(G).

$$\begin{aligned} x(\delta(V_i)) &\geq con(V_i), && \text{for } i = 1, \dots, p, \\ -x(e) &\geq -1, && \text{for all } e \in F, \\ x(e) &\geq 0, && \text{for all } e \in \delta(V_0) \setminus F. \end{aligned}$$

By summing these inequalities we obtain

$$2x(\delta(\pi) \setminus F) \geq \sum_{i=1}^{p} con(V_i) - |F|.$$

By dividing by 2, and rounding up the right-hand side, we obtain

$$x(\delta(\pi) \setminus F) \geq \left\lceil \frac{\sum_{i=1}^{p} con(V_i) - |F|}{2} \right\rceil. \tag{12}$$

Inequalities of type (12) will be also called *F-partition inequalities*

Claim 1. $\sum_{i=1}^{p} con(V_i) = 2p - p_1(\pi) + p_3(\pi)$.

Proof. $\sum_{i=1}^{p} con(V_i) = p_1(\pi) + 2p_2(\pi) + 3p_3(\pi) = 2p_1(\pi) + 2p_2(\pi) + 2p_3(\pi) + p_3(\pi) - p_1(\pi)$. As $p_1(\pi) + p_2(\pi) + p_3(\pi) = p$, the statement follows. □

By Claim 1, inequality (12) can be written as

$$x(\delta(\pi) \setminus F) \geq \left\lceil \frac{2p - p_1(\pi) + p_3(\pi) - |F|}{2} \right\rceil. \tag{13}$$

The following theorem gives necessary conditions for inequalities of type (13) to define facets.

Recall that a subset of nodes in a graph is called a *stable set* if no edge exists between any pair of its nodes.

Theorem 4. *Suppose that $con(V_i) \geq 1$ for $i = 1, \ldots, p$. Then inequality (13) defines a facet for the ESNDP(G) only if*

1. $\sum_{i=1}^{p} con(V_i)$ *and* $|F|$ *have different parity.*
2. $G(V_i)$ *is connected for* $i = 1, \ldots, p$.
3. *If* $G(V_i)$, $i \in \{1, \ldots, p\}$, *is not 2-edge connected (and connected), then any partition* V_i^1, V_i^2 *of* V_i *such that* $\left|[V_i^1, V_i^2]\right| = 1$ *and* $con(V_i^l) = con(V_i)$, $l \in \{1, 2\}$, *then* $con(V_i^{l'}) \leq 1$ *where* $l' = \{1, 2\} \setminus \{l\}$.
4. *If* $\left|\delta(V_i)\right| = 2$ *for* $i \in \{1, \ldots, p\}$, *then* $con(V_i) \leq 1$.
5. *If* $\left|F \cap \delta(V_i)\right| = 1$ *for* $i = 1, \ldots, p$, *then the set* $\{V_i \mid con(V_i) = 1\}$ *is a stable set.*

Proof. See Appendix A.2. □

Next, we describe sufficient conditions for the F-partition inequalities to define facets.

Theorem 5. *Let* $\pi = (V_0, V_1, \ldots, V_p)$ *be a partition of* V *such that* $con(V_1) = 1, con(V_2) = \cdots = con(V_{p_2(\pi)+1}) = 2$ *and* $con(V_{p_2(\pi)+2}) = \cdots = con(V_p) = 3$. *Suppose that*

1. $G(V_1)$ *is connected.*
2. $G(V_i)$ *is 3-edge connected for* $i = 0, 2, \ldots, p_2(\pi) + 1$ *and* $G(V_i)$ *is 4-edge connected for* $i = p_2(\pi) + 2, \ldots, p$.
3. $\left|[V_i, V_{i+1}]\right| \geq 1$ *for* $i = 2, \ldots, p_2(\pi)$,
   $\left|[V_i, V_{i+1}]\right| \geq 2$ *for* $i = p_2(\pi) + 2, \ldots, p_2(\pi) + p_3(\pi)$.
4. $\left|(F \cap \delta(V_i)\right| = 1$ *for* $i = 1, \ldots, p$.
5. $\sum_{i=1}^{p} con(V_i)$ *and* $|F|$ *have different parities.*
6. G_π *is complete.*

Then inequality (13) defines a facet of ESNDP(G).

Proof. First, note that $p = p_2(\pi) + p_3(\pi) + 1$. Also, note that by the conditions it follows that there is no essential edge, and therefore ESNDP(G) is fully dimensional.

Let $ax \geq \alpha$ denote inequality (13), and let $bx \geq \beta$ be a facet-defining inequality of ESNDP(G) such that

$$\{x \in \text{ESNDP}(G) \mid ax = \alpha\} \subseteq \{x \in \text{ESNDP}(G) \mid bx = \beta\}.$$

We will show that there is a scalar ρ such that $b = \rho a$.

To this end, we first show that $b(e) = b(e')$ for all $e, e' \in \delta(\pi) \setminus F$. Let $F = \{f_1, \ldots, f_p\}$, where $f_i \in \delta(V_i)$ for $i = 1, \ldots, p$.

By Condition (3), there is an edge, say e_i, in $[V_i, V_{i+1}]$ for $i = 2, 3, \ldots, p \,(\mathrm{mod}\, p)$.

Also, we claim that $p_2(\pi)$ is odd. In fact, we have that $p = p_1(\pi) + p_2(\pi) + p_3(\pi)$, and by Claim 1, $\sum_{i=1}^{p} con(V_i) = 2p - p_1(\pi) + p_3(\pi)$. As $|F| = p$ and $p_1(\pi) = 1$, this yields

$$\begin{aligned}\sum_{i=1}^{p} con(V_i) - |F| &= 2p - 1 + p_3(\pi) - p\\ &= p - 1 + p_3(\pi)\\ &= 1 + p_2(\pi) + p_3(\pi) - 1 + p_3(\pi)\\ &= 2p_3(\pi) + p_2(\pi).\end{aligned}$$

As $\sum_{i=1}^{p} con(V_i)$ and $|F|$ have different parities and therefore their difference should be odd, we obtain that $p_2(\pi)$ is odd. Thus by (A4), inequality (13) can be written as

$$x(\delta(\pi) \setminus F) \geq p_3(\pi) + \frac{p_2(\pi) + 1}{2}.$$

Consider the edge sets

$$S_0 = \{e_2, e_4, \ldots, e_{p_2(\pi)-1}, e_{p_2(\pi)}\} \cup \{e_{p_2(\pi)+1}, e_{p_2(\pi)+2}, \ldots, e_{p_2(\pi)+p_3(\pi)+1}\} \cup F \cup \bigcup_{i=0}^{p} E(V_i),$$

(see Fig. 3).

$$S_1 = (S_0 \setminus \{e_p\}) \cup \{e\},$$

where e is an edge of $\delta(V_p) \setminus \{e_{p-1}, e_p, f_p\}$.

It is easy to see that S_0 and S_1 are solutions of ESNDP and $ax^{S_0} = ax^{S_1} = \alpha$. Hence, $bx^{S_0} = bx^{S_1} = \beta$, implying that $b(e_p) = b(e)$. By symmetry, it follows that

$$b(e) = \rho \qquad \text{for all } e \in \delta(V_p) \setminus \{e_{p-1}, f_p\} \text{ for some scalar } \rho. \tag{14}$$

Now let

$$S_2 = (S_0 \setminus \{e_{p-1}\} \cup \{g\}),$$

where g is an edge of $[V_{p-1}, V_p] \setminus \{e_{p-1}\}$. Note that by Condition (3) such an edge exists. Trivially, S_2 is a solution of ESNDP(G) and $ax^{S_2} = \alpha$. Hence $bx^{S_2} = \beta$. Therefore, $0 = bx^{S_0} - bx^{S_2} = b(e_{p-1}) - b(g)$. As by (14), $b(g) = \rho$, it follows that $b(e_{p-1}) = \rho$ and, consequently, $b(e) = \rho$ for all $e \in \delta(V_p) \setminus \{f_p\}$. Since G is complete, any set V_j, $p_2(\pi) + 1 \leq j \leq p - 1$, can play the same role as V_p. Moreover, $\delta(V_j) \cap \delta(V_{j+1}) \neq \emptyset$ ($e_j \in \delta(V_j) \cap \delta(V_{j+1})$). Hence, by symmetry, we obtain that

$$b(e) = \rho \qquad \text{for all } e \in \delta(V_i) \setminus \{f_i\}, \text{ for } i = p_2(\pi) + 1, \ldots, p. \tag{15}$$

Now consider the solution

$$S_3 = (S_0 \setminus \{e_{p-1}, e_p, e_2\}) \cup \{g_1, g_2, g_3\},$$

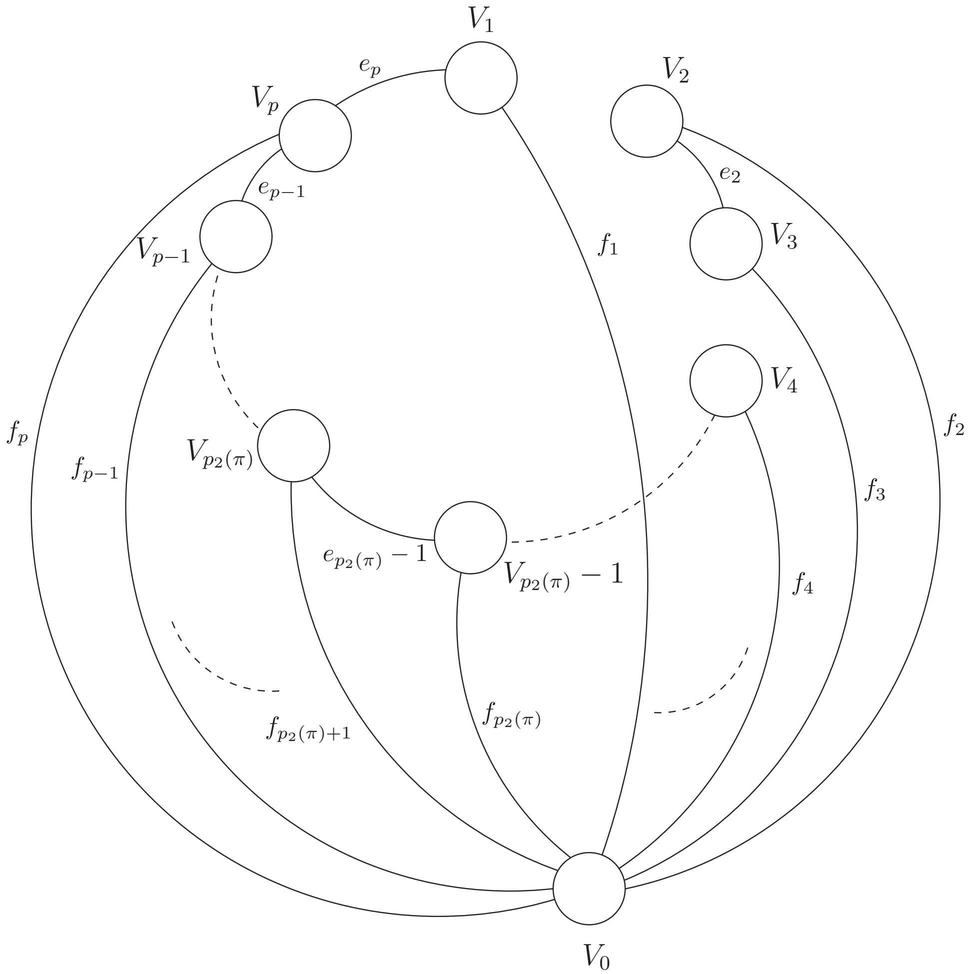

Fig. 3. Solution S_0.

where $g_i \in [V_p, V_i]$, $i = 2, 3$ and $g_3 \in [V_{p-1}, V_1]$. Since S_3 is a solution of ESNDP(G) and $ax^{S_0} = ax^{S_3} = \alpha$, this implies that $bx^{S_0} = bx^{S_3} = \beta$ and therefore $b(e_2) + b(e_p) + b(e_{p-1}) = b(g_1) + b(g_2) + b(g_3)$. As by (15), $b(e_p) = b(e_{p-1}) = b(g_1) = b(g_2) = b(g_3) = \rho$, we have that $b(e_2) = \rho$.

As any edge of $[V_2, V_i]$, $3 \le i \le p_2(\pi)$ can play the same role as e_2, we obtain, by symmetry, that $b(e) = \rho$ for all $e \in \delta(V_2) \setminus \{f_2\}$. And again, by symmetry, it follows that

$$b(e) = \rho \qquad \text{for all } e \in \delta(V_i) \setminus \{f_i\} \text{ for } i = 1, \ldots, p_2(\pi). \tag{16}$$

Next, we show that $b(f) = 0$ for all $f \in F$.

Consider the solutions

$$\begin{aligned} S_5 = &\quad (S_0 \setminus \{e_p\}) \cup \{g_2\}, \\ S_5' = &\quad S_5 \setminus \{f_2\}, \end{aligned}$$

where $g_2 \in [V_p, V_2]$.

Both sets S_5 and S'_5 are solutions of ESNDP(G) and $ax^{S_5} = ax^{S'_5} = \alpha$. This yields $0 = bx^{S_5} - bx^{S'_5} = b(f_2)$. And by symmetry, it follows that $b(f_i) = 0$ for all $i = 3, \dots, p_2(\pi)$.

Also, consider the sets

$$S_6 = \quad (S_0 \setminus \{e_p, e_2\}) \cup \{g_2, g_3\},$$
$$S'_6 = \quad S_5 \setminus \{f_p\},$$

where $g_i \in [V_p, V_i]$, $i = 2, 3$.

As S_6 and S'_6 are solutions of ESNDP(G), we have that $0 = bx^{S_6} - bx^{S'_6} = b(f_p)$. As before, we obtain, by symmetry, that $b(f_i) = 0$ for all $i = p_2(\pi) + 1, \dots, p - 1$.

Moreover, let $S'_3 = S_3 \setminus \{f_1\}$, where S_3 is the edge set considered above.

As S'_3 is also a solution of ESNDP(G), and $ax^{S'_3} = \alpha$, we get $b(f_1) = 0$. Thus, we have

$$b(f_i) = 0 \qquad \text{for } i = 1, \dots, p. \tag{17}$$

By Condition (2), it follows that the solution

$$S_7 = \quad S_0 \setminus \{e\} \qquad \text{for } e \in E(V_i) \text{ and } i \in \{0, 1, \dots, p\},$$

remains feasible for the ESNDP. As $ax^{S_7} = \alpha$, we have $0 = bx^{S_0} - bx^{S_7} = b(e)$. Hence

$$b(e) = 0 \qquad \text{for all } e \in E(V_i),\ i = 0, 1, \dots, p. \tag{18}$$

By (16)–(18), we obtain that

$$b(e) = \rho \qquad \text{for all } e \in \delta(pi) \setminus F,$$
$$b(e) = 0 \qquad \text{for all } e \in F \cup \bigcup_{i=0}^{p} E(V_i).$$

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

3.3. Partition inequalities

In this section, we describe a further class of valid inequalities induced by partitions of the underlying graph.

Let $G = (V, E)$ be a graph and $r \in \mathbb{N}^V$ a connectivity-type vector. Let $\pi = (V_1, \dots, V_p)$ be a partition such that $V_i \neq \emptyset$ and $r(V_i) \geq 1$ for $i = 1, \dots, p$. Let $I_1 = \{i \mid con(V_i) = 1\}$ and $I'_1 = \{i \mid con(V_i) \geq 2\}$. Kerivin (2000) shows that the following inequality is valid for the survivable network design problem:

$$x(\delta(\pi)) \ \geq \ \begin{cases} p - 1 & \text{if } I'_1 = \emptyset, \\ \left\lceil \frac{1}{2} \sum_{i \in I'_1} con(V_i) \right\rceil + |I_1| & \text{if not.} \end{cases} \tag{19}$$

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

Lemma 2. *Let $I_i = \{j \mid con(V_j) = i\}$ for $i = 2, 3$. If $r \in \{0, 1, 2, 3\}^V$, then inequality (19) can be written as*

$$x(\delta(\pi)) \geq \begin{cases} p-1 & \text{if } I_2 = \emptyset = I_3, \\ p & \text{if } I_2 \neq \emptyset, I_3 = \emptyset, \\ p + \left\lceil \frac{|I_3|}{2} \right\rceil & \text{if } I_3 \neq \emptyset. \end{cases} \tag{20}$$

Proof. If $I_2 = \emptyset = I_3$, then inequality (19) reduces to

$$x(\delta(\pi)) \geq p - 1. \tag{21}$$

Here an edge set is a solution of ESNDP if it connects the sets $V_1, \ldots, V_p$, that is to say, if it contains a spanning tree on $V_1, \ldots, V_p$. For this, we need at least $p-1$ edges.

If $I_3 = \emptyset$ and $I_2 \neq \emptyset$ ($|I_2| \geq 2$), then inequality (19) becomes

$$\begin{aligned} x(\delta(\pi)) &\geq \left\lceil \frac{1}{2} \sum_{i \in I_2} con(V_i) \right\rceil + |I_1| \\ &= |I_2| + |I_1| \\ &= p. \end{aligned}$$

Since $|I_2| \geq 2$, a solution of ESNDP, in addition to connecting the sets $V_1, \ldots, V_p$, it must contain at least two edge-disjoint paths between two sets among $V_1, \ldots, V_p$. Thus, it must contain at least p edges.

Now suppose that $I_3 \neq \emptyset$. We will distinguish two cases.

Case 1: $|I_3|$ even.

We have

$$\begin{aligned} x(\delta(\pi)) &\geq \left\lceil \frac{1}{2} \sum_{i \in I_2 \cup I_3} con(V_i) \right\rceil + |I_1| \\ &= |I_2| + \left\lceil \tfrac{3}{2} |I_3| \right\rceil + |I_1| \\ &= |I_2| + \tfrac{3}{2} |I_3| + |I_1| \text{ (since } |I_3| \text{ is even)} \\ &= |I_2| + |I_3| + \tfrac{|I_3|}{2} + |I_1| \\ &= p + \left\lceil \frac{|I_3|}{2} \right\rceil, \end{aligned}$$

as $p = |I_1| + |I_2| + |I_3|$.

Case 2: $|I_3|$ odd.

Thus

$$\begin{aligned} x(\delta(\pi)) &\geq \left\lceil \frac{1}{2} \sum_{i \in I_2 \cup I_3} con(V_i) \right\rceil + |I_1| \\ &= |I_2| + \left\lceil \tfrac{3}{2}|I_3| \right\rceil + |I_1| \\ &= |I_2| + |I_3| + \left\lceil \tfrac{|I_3|}{2} \right\rceil + |I_1| \\ &= p + \left\lceil \tfrac{|I_3|}{2} \right\rceil . \end{aligned}$$

□

In what follows we give necessary conditions for inequality (20) to define a facet for ESNDP(G) when $I_3 \neq \emptyset$.

Theorem 6. *Suppose $I_3 \neq \emptyset$. Then inequality (20) defines a facet of ESNDP(G) only if*

1. *$|I_3|$ is odd.*
2. *$G(V_i)$ is 2-edge connected for $i = 1, \ldots, p$.*

Proof.

1. Trivial.
2. Suppose first, on the contrary, that $G(V_i)$ is not connected for some $i \in \{1, \ldots, p\}$. Then there is a partition V_i^1, V_i^2 such that $[V_i^1, V_i^2] = \emptyset$. Let $\pi' = (V_1', \ldots, V_{p+1}')$ be the partition given by

$$\begin{cases} V_l' = V_l, & l = 1, \ldots, i-1, \\ V_i' = V_i^1, & \\ V_{i+1}' = V_i^2, & \\ V_l' = V_{l-1}, & l = i+1, \ldots, p+1. \end{cases}$$

Inequality (20), induced by π', looks as

$$x(\delta(\pi')) \geq p + 1 + \left\lceil \frac{|I_3'|}{2} \right\rceil , \tag{22}$$

where I_3' is the set of V_i' with connectivity 3.
As $\delta(\pi) = \delta(\pi')$, clearly, inequality (20) (when $I_3 \neq \emptyset$) is dominated by (22), and hence cannot be facet defining.
Now suppose that there is a set V_i such that $G(V_i)$ is connected but not 2-edge connected. Then there is a partition V_i^1, V_i^2 of V_i and an edge e such that $[V_i^1, V_i^2] = \{e\}$.
Let π' be the partition as defined above. It is easy to see that inequality (20) can be obtained as the sum of inequality (22) and the trivial inequality $-x(e) \geq -1$. Therefore, it cannot define a facet. □

Now, we will introduce a new class of valid inequalities.

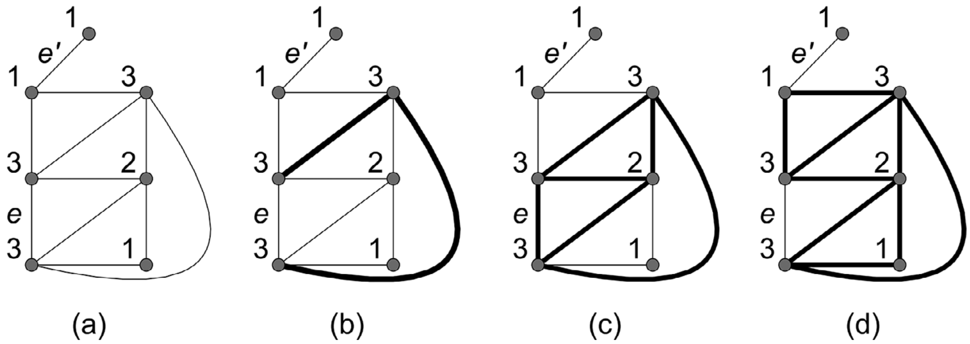

Fig. 4. Graph G, T, $L_e^T \cup \{e\}$ and $\bar{L}_e^T$, with connectivity types on the nodes.

3.4. T-completion inequalities

Consider a subset of edges $T \subseteq E$, $T \neq \emptyset$. Let $k = \min\{r_v\}$, $v \in V(T)$. For edge $e \in E \setminus T$, let $L_e^T, \bar{L}_e^T \subseteq E \setminus (T \cup \{e\})$ be two sets, defined as follows:

- L_e^T is the set of edges of a minimum size such that the graph induced by $T \cup \{e\} \cup L_e^T$ has k-edge-disjoint paths between every pair of nodes in $V(T)$.
- $\bar{L}_e^T$ is the set of edges of a minimum size such that the graph induced by $T \cup \bar{L}_e^T$ has k-edge-disjoint paths between every pair of nodes in $V(T)$.

In what follows, we suppose that edge e is not essential for the problem, i.e., $(V, E \setminus \{e\})$ is still a feasible solution. Otherwise, $\bar{L}_e^T$ would not have sense.

In order to illustrate L_e^T and $\bar{L}_e^T$, consider graph G of Fig. 4a, where the number on each node is the connectivity type of the node. The graph of Fig. 4b shows a tree T given by the bold edges and an edge e. Observe that $V(T)$ only contains nodes of connectivity type 3. Hence $k = 3$. Figure 4c (resp. Fig. 4d) displays $L_e^T \cup T \cup \{e\}$ (resp. $\bar{L}_e^T \cup T$) with the bold edges.

Remark 1. For all $T \subseteq V$ and $e \in E \setminus T$, $|\bar{L}_e^T| \geq |L_e^T|$.

Proposition 1. *The following inequality*

$$\sum_{f \in E \setminus (T \cup \{e\})} x(f) + (|\bar{L}_e^T| - |L_e^T|)x(e) \geq |\bar{L}_e^T| \tag{23}$$

is valid for ESNDP(G).

Proof. Let S be a solution of the ESNDP(G). If $e \notin S$ (resp. $e \in S$), then $x^S(e) = 0$ (resp. $x^S(e) = 1$), and thus $\sum_{f \in E \setminus (T \cup \{e\})} x^S(f) \geq |\bar{L}_e^T|$ (resp. $\sum_{f \in E \setminus (T \cup \{e\})} x^S(f) \geq |L_e^T|$). Both inequalities come from the minimality of L_e^T and $\bar{L}_e^T$, respectively. Hence the statement follows. □

For the example of Fig. 4, we have $|L_e^T| = 3$ and $|\bar{L}_e^T| = 7$. Hence, the corresponding inequality is given by

$$\sum_{f \in E \setminus (T \cup \{e\})} x(f) + 4x(e) \geq 7.$$

Inequalities of type (23) will be called *T-completion inequalities*.

An edge $e' \in E \setminus (T \cup \{e\})$ is called *redundant* for e and T, if $|L_e^T| = |L_e^{T \cup \{e'\}}|$ and $|\bar{L}_e^T| = |\bar{L}_e^{T \cup \{e'\}}|$. In Fig. 4, arc e' is redundant for e and T. Indeed, we can easily see that $|L_e^T| = |L_e^{T \cup \{e'\}}| = 3$ and $|\bar{L}_e^T| = |\bar{L}_e^{T \cup \{e'\}}| = 7$.

Proposition 2. *Inequality (23), associated with $T \subseteq E$ and edge $e \in E \setminus T$, defines a facet of ESNDP(G) only if every edge of $E \setminus (T \cup \{e\})$ is not redundant for e and T.*

Proof. Suppose that there exists an edge $e' \in E \setminus (T \cup \{e\})$ that is redundant for e and T. Then, inequality (23), associated with T and e, can be obtained by summing inequality (23), associated with T' and e, and $x(e') \geq 0$, where $T' = T \cup \{e'\}$. □

4. Reduction operations

In this section, we describe some reduction operations that permit to reduce the size of the graph with respect to a given fractional extreme point of the linear relaxation of the relaxed polytope, while the violated inequalities are preserved in the new graph. As it turns out, these will be useful in the branch-and-cut algorithm.

Consider the linear relaxation of the ESNDP(G), P(G) given by

$$\mathrm{P}(G) \begin{cases} x(\delta(W)) \geq con(W) & \text{for all } \emptyset \neq W \subset V, \\ 0 \leq x(e) \leq 1 & \text{for all } e \in E. \end{cases}$$

Let $\overline{x}$ be an extreme point of P(G). Then $\overline{x}$ is the unique solution of a system $S(\overline{x})$ of the form

$$S(\overline{x}) \begin{cases} x(e) = 0 & \text{for all } e \in E_0(\overline{x}), \\ x(e) = 1 & \text{for all } e \in E_1(\overline{x}), \\ x(\delta(W)) = con(W) & \text{for all } \delta(W) \in \mathcal{C}(\overline{x}), \end{cases}$$

where $E_0(\overline{x})$ (resp. $E_1(\overline{x})$) is the set of edges e such that $\overline{x}(e) = 0$ (resp. $\overline{x}(e) = 1$) and $\mathcal{C}(\overline{x})$ is a family of cuts which are tight for $\overline{x}$ such that

$$|E_0(\overline{x})| + |E_1(\overline{x})| + |\mathcal{C}(\overline{x})| = |E|.$$

Consider the following operations:

- θ_1: Remove an edge e with $\overline{x}(e) = 0$.

- θ_2: Contract an edge $e_1 = uv_1$ such that one of its extremities, say u, is of degree 2, $r(u) = 2$, $r(v_1) = 2$, and $\overline{x}(e_1) = \overline{x}(e_2) = 1$, where $\delta(u) = \{e_1, e_2\}$. Set $r(w) = 2$, where w is the node obtained by contracting e_1 (see Fig. A2).
- θ_3: Contract a subset of nodes W such that $G(W)$ is k-edge connected, $r(v) = k$, for all $v \in W$ and $\overline{x}(e) = 1$ for all $e \in E(W)$, for $k = 1, 2, 3$. Set $r(w) = k$ where w is the node resulting from the contraction.

We have the following theorem.

Theorem 7. *Let $G' = (V', E')$ be the graph obtained by repeated applications of operations $\theta_1, \theta_2, \theta_3$, and $\overline{x}'$ the restriction of $\overline{x}$ on E'. Then, $\overline{x}'$ is an extreme point of $P(G')$.*

Proof. See Appendix A.3. □

As an algorithmic consequence of operations $\theta_1, \theta_2, \theta_3$, we have the following result which is not hard to be seen as true.

Theorem 8. *If a partition-based inequality (cut, partition, F-partition, or SP-partition inequality) is violated by $\overline{x}$ in G, then the same inequality is violated by $\overline{x}'$ in G′.*

The previous theorem is of great algorithmic interest. Indeed, one can apply $\theta_1, \theta_2, \theta_3$ in a pre-processing procedure before any separation phase in a branch-and-cut algorithm. The separation routines can then be applied in the reduced graph $G' = (V', E')$ with respect to the solution $\overline{x}'$. Any violated inequality found in G', with respect to $\overline{x}$, can be straightforwardly lifted to a violated inequality of the same type in G with respect to $\overline{x}$.

5. Branch-and-cut algorithm

In this section, we describe our branch-and-cut algorithm for the ESNDP when $r(v) \in \{1, 2, 3\}$ for all $v \in V$. We have not considered connectivity 0 in the experiments since all the valid inequalities are partition-based inequalities, and every set V_i of the underlying partition is such that $con(V_i) \geq 1$. Moreover, in the majority of cases, this may add a nonproportional difficulty to the instances making them harder to solve within the fixed time limit. Our aim is particularly to test the efficiency of the valid inequalities. The algorithm will be based on the theoretical results presented in the previous sections. It will use the reduction operations in a preprocessing phase before every separation procedure.

Given a solution $\overline{x}$ of ESNDP(G), let $\overline{x}'$ and $G' = (V', E')$ be a solution and a graph obtained by means of operations $\theta_1, \theta_2, \theta_3$ from $\overline{x}$. As shown by Theorem 7, $\overline{x}'$ is an extreme point of ESNDP(G') if and only if $\overline{x}$ is an extreme point of ESNDP(G). Moreover, we have the following result which can be easily seen to be true.

Proposition 3. *Let $\pi' = (V'_1, \ldots, V'_p)$ (resp. $\pi' = (V'_0, V'_1, \ldots, V'_p)$) be a partition of V inducing a partition (SP-partition) (resp. F-partition) inequality $a'x \geq \alpha'$ valid for ESNDP(G'). Then, the* *partition $\pi = (V_1, \ldots, V_p)$ (resp. $\pi = (V_0, V_1, \ldots, V_p)$), obtained by expanding the sets V_i' of π', induces a valid inequality $ax \geq \alpha'$ of the ESNDP(G) such that*

$$a(e) = \begin{cases} a'(e), & \text{if } e \in E', \\ 1, & \text{if } e \in (E \setminus E') \cap \delta(\pi), \\ 0, & \text{if not}. \end{cases} \tag{24}$$

Moreover, $a'x \geq \alpha'$ is violated by $\overline{x}'$ if and only if $ax \geq \alpha'$ is violated by $\overline{x}$.

Proof. First observe that $r(\pi') = r(\pi)$, and $p_i(\pi') = p_i(\pi)$ for $i = 1, 2, 3$. Moreover, the edges of $(E \setminus E') \cap \delta(\pi)$ are those edges of $\delta(\pi)$ with $\overline{x}(e) = 0$. Then clearly, $ax \geq \alpha'$ is valid for the ESNDP(G). In addition, as $x(\delta(\pi)) = x(\delta(\pi)')$, the statement follows. □

By Proposition 3, given a fractional solution $\overline{x}$ of the ESNDP(G), not necessarily an extreme point, one can apply operations $\theta_1, \theta_2, \theta_3$ with respect to $\overline{x}$ until no more operations could be performed.

Note that operations $\theta_1, \theta_2, \theta_3$ can be implemented in polynomial time and any order. Then separating the partition, SP-partition, and F-partition inequalities for $\overline{x}$ in G reduces to separating these inequalities for $\overline{x}'$ in G'. By Proposition 3, the obtained violated inequalities can be lifted to violated inequalities for $\overline{x}$ in G.

5.1. Framework of the algorithm

The optimization starts with the following linear program given by the cut inequalities induced by the nodes of the graph together with the trivial inequalities,

$$\begin{aligned} &\min \sum_{e \in E} c(e)x(e) \\ &x(\delta(v)) \geq con(v) \quad \text{for all } v \in V, \\ &0 \leq x(e) \leq 1 \quad \text{for all } e \in E. \end{aligned} \tag{25}$$

Here $con(v)$ stands for $con(\{v\})$.

If the optimal solution $\overline{x} \in \mathbb{R}^E$ of this linear program is integer and satisfies the cut inequalities (2), then it is optimal for the problem. If not, then one needs to address the separation problem for the different classes of inequalities in order to generate further valid inequalities violated by $\overline{x}$.

Recall that the separation problem for a class $\mathcal{C}$ of inequalities and a solution $\overline{x}$ consists of determining whether $\overline{x}$ satisfies all the inequalities in $\mathcal{C}$ and if not finding an inequality of $\mathcal{C}$ violated by $\overline{x}$.

In our branch-and-cut algorithm, we use the families of inequalities described in the previous section. Preliminary experimentation showed the efficiency of proceeding for the separation in the following order:

1. Cut inequalities,
2. SP-partition inequalities,

3. F-partition inequalities,
4. Partition inequalities.

We go from one class to the next one only when no more violated inequalities in the class can be generated. Several inequalities can be added at the same time.

We remark that the separations are performed in the reduced graph obtained by the reduction operations $\theta_1, \theta_2, \theta_3$, with respect to the current fractional solution. Also, remark that the inequalities are global, that is to say, they are valid in the whole branch-and-cut tree.

5.2. Separation algorithms

Now we present the separation routines used in the algorithm.

5.2.1. Cut inequalities separation

The separation of the cut inequalities reduces to finding a minimum cut in G with respect to the weights $\overline{x}(e)$, $e \in E$. Moreover, computing a minimum cut can be done in polynomial time using the Gusfield algorithm (Gusfield, 1990). This processes the so-called Gomory–Hu tree (Gomory and Hu, 1961). Any minimum cut separating two nodes s and t in the tree is also a minimum cut separating s and t in G. So we separate the cut inequalities exactly. We also separate them in the reduced graph G'.

5.2.2. SP-partition inequalities separation

Now we discuss the separation problem of the SP-partition inequalities. From Baïou et al. (2000), if the underlying graph is series–parallel, these inequalities can be separated in polynomial time. Moreover, if $r \in \{1, 2\}^V$, inequalities (7) reduce to

$$x(\delta(\pi)) \geq \begin{cases} p, & \text{if } r(\pi) = 2, \\ p - 1, & \text{if } r(\pi) = 1. \end{cases} \tag{26}$$

Inequalities (26) are nothing but the partition inequalities (20) when $r \in \{1, 2\}^V$. Then, they can be separated in polynomial time using Kerivin and Mahjoub's algorithm (Kerivin and Mahjoub, 2002) when the underlying graph is series–parallel. Both Baïou et al. (2000) and Kerivin and Mahjoub (2002) algorithms reduce the separation problem to the minimization of a submodular function which can be done in polynomial time.

In our branch-and-cut algorithm, we used heuristic procedures for the separation of the SP-partition inequalities. Our instances consider both uniform and nonuniform cases. For the uniform case, we use the heuristics developed by Bendali et al. (2010). For the nonuniform instances, we devised a heuristic based on Theorems 2 and 3. The aim is to find a partition $\pi = (V_1, \ldots, V_p)$ inducing a series–parallel graph with $|[V_i, V_{i+1}]| \geq 2$ (resp. $|[V_i, V_{i+1}]| \geq 1$) for every $i \in \{1, \ldots, p\}$ such that $con(V_i) = con(V_{i+1}) = 3$ (resp. either $con(V_i) \neq 3$ or $con(V_{i+1}) \neq 3$ or both), where the indices are taken modulo p and the graph $G(\pi)$ is 2-node connected. For this, we consider the graph $G' = (V', E')$ obtained using the reduction operations $\theta_1, \theta_2, \theta_3$. We look for a path

$P = (v'_1, \dots, v'_q)$ going through nodes of connectivity 3 such that $|[v'_i, v'_{i+1}]| \geq 2$ for $i = 1, \dots, q-1$. We also look for the longest path Q, in terms of the number of edges, between v'_q and v'_1, not going through any other node of P. Let $v'_q, v'_{q+1}, \dots, v'_p, v'_1$ be the sequence of nodes of Q. Let V_i (resp. V_1, resp. V_p) be the subset of V obtained by expanding $v'_i, i = 2, \dots, p-1$ (resp. the set $\{v'_1\} \cup \{v'_j \in V' \setminus P \,|\, r(v'_j) = 3\}$, resp. the set $\{v'_p\} \cup \{v'_j \in V' \setminus (P \cup Q) \,|\, r(v'_j) \leq 2\}$). Afterwards, we verify whether the graph $G(\pi)$, where $\pi = (V_1, \dots, V_p)$, is series–parallel and 2-node connected. Finally, we check whether inequality (7) induced by $\pi = (V_1, \dots, V_p)$ is violated by $\overline{x}$. If either $G(\pi)$ is not series–parallel or not 2-node connected, or inequality (7) is not violated by $\overline{x}$, then we repeat again the procedure to look for other paths. The new paths are computed in the graph obtained by removing the nodes involved in the previous paths. This process continues until either a violated inequality of type (7) is obtained or it is not possible to find other paths.

5.2.3. F-partition inequalities separation

Now we turn our attention to the separation of the F-partition inequalities (12). These inequalities can be separated in polynomial time in the uniform case when F is fixed. Their separation's complexity when F is not fixed, even in the uniform case, is still an open question. For our purpose, in the uniform case, we use the heuristic developed by Bendali et al. (2010) to separate them. In the nonuniform case, we use the heuristic below based on the following proposition.

Proposition 4. *Consider a partition $\pi = (V_0, V_1, \dots, V_p)$ of V and let $F \subseteq \delta(V_0)$. Suppose that $\overline{x}(\delta(V_i)) = con(V_i)$, $i = 1, \dots, p$. Then inequality (12) is violated by $\overline{x}$ if*

$$\overline{x}(\delta(V_0) \setminus F) + |F| - x(F) < 1. \tag{27}$$

Proof. As $\overline{x}(\delta(V_i)) = con(V_i)$, $i = 1, \dots, p$, by summing these equations, we have

$$\sum_{i=1}^{p} x(\delta(V_i)) = 2x(\delta(\pi)) + x(\delta(V_0)) = \sum_{i=1}^{p} con(V_i).$$

By adding this equation to inequality (27), the result follows. □

The heuristic starts by looking for nodes v in $G' = (V', E')$ with $\overline{x}(\delta(v)) = con(v)$. Let $v'_1, \dots, v'_p$ be the obtained nodes. We let V_i (resp. V_0) be the node set obtained by expanding $v'_i, i = 1, \dots, p$ (resp. $V' \setminus \{v'_1, \dots, v'_p\}$). We choose F among the edges of $\delta(V_0)$, with high values and such that $|F|$ and $\sum_{i=1}^{p} con(V_i)$ have different parities. Then we verify whether inequality (27) is satisfied. If so, then inequality (12) induced by $(V_0, V_1, \dots, V_p)$ and F is violated by $\overline{x}$.

5.2.4. Partition inequalities separation

Finally, we discuss the separation problem for the partition inequalities (20). Grötschel et al. (1992a) show that the separation problem for the partition inequalities when $r \in \{0, 1, 2\}^V$ is NP-hard. In what follows, we discuss the case when $r \in \{0, 1, 2, 3\}^V$.

The separation problem in this case can be handled in polynomial time in some special cases.

If $I_2 = \emptyset = I_3$, then the corresponding partition inequality (20) is equivalent to (21). Here, one can shrink the nodes of types 2 and 3 in a node w and assign to w the connectivity type 1. Then the separation of the inequalities of type (20) in G is equivalent to their separation in the new graph where every node has a connectivity type 1. To do this, one can use either Cunningham's algorithm (Cunningham, 1985) or Barahona's algorithm (Barahona, 1992). Both algorithms give the most violated inequality if there is any. Cunningham's algorithm runs in $O(|E|)^3$, whereas Barahona's algorithm runs in $O(|V|)^3$. For our purpose, we have used Barahona's algorithm. This reduces the problem to $|V|$ minimum cut problems.

If $I_2 \neq \emptyset$ and $I_3 = \emptyset$, then the partition inequality is given by

$$x(\delta(\pi)) \geq p. \tag{28}$$

Here one can shrink the nodes of type 3 to one node and give the new node the connectivity type 2 or 3, and then solve the separation problem for the inequalities of type (28) in the resulting graph.

As mentioned before, Kerivin and Mahjoub (2002) show that the separation problem for the inequalities of type (28) reduces to the minimization of a submodular function and can then be solved in polynomial time.

If $I_3 \neq \emptyset$, and the number of nodes of type 3 is 2, say v_1, v_2, then the partition inequality when v_1 and v_2 are in different sets of the partition can be written as

$$x(\delta(\pi)) \geq p + 1. \tag{29}$$

The separation problem of the partition inequalities of type (29) can be solved in polynomial time along the same line as inequalities (28).

If the number of nodes of connectivity type 3 is ≥ 3, one can separate in polynomial time inequalities (29) where all the nodes of type 3 except one are in a set of partitions and the remaining nodes in another set. This reduces to shrinking the nodes of type 3 except one each time, giving to the new node connectivity type 3, and solving the separation problem in the resulting graph where the two nodes of type 3 are in different sets of the partition.

Since the previous exact techniques are time-consuming, in the case where either $I_2 \neq \emptyset$ or $I_3 \neq \emptyset$ or both, we used a heuristic procedure for separating the partition inequalities in this case.

The heuristic is based on the following observation: Given a partition $(V_1, \dots, V_p)$ of V, if $\overline{x}(\delta(V_i)) = con(V_i)$ for $i = 1, \dots, p$, then inequality (20) is violated by $\overline{x}$. So the idea of the heuristic is to determine precisely partitions in which all the cuts induced by its elements are tight for $\overline{x}$. To this end, we look in the graph $G' = (V', E')$ for a Gomory–Hu tree T with respect to $\overline{x}$ (Gomory and Hu, 1961). Afterwards, we look for disjoint subsets $W_1, \dots, W_q$ tight for $\overline{x}$. Let $w_{q+1}, \dots, w_p$ be the nodes of $V' \setminus \left(\bigcup_{i=1}^{q} W_i\right)$. Then we consider the partition $(V_1, \dots, V_p)$, where V_i is the node set obtained by expanding W_i (resp. $\{w_i\}$) for $i = 1, \dots, q$ (resp. $i = q+1, \dots, p$). Finally, we verify whether or not the corresponding inequality (20) is violated by $\overline{x}$.

5.2.5. T-completion inequalities separation

Given $x^* \in [0, 1]^E$, the separation problem for (23) reduces to finding a subset of edges $T \subseteq E$ in G and an edge $e \in E \setminus T$ such that $\sum_{f \in E \setminus (T \cup \{e\})} x^*(f) + (|\bar{L}_e^T| - |L_e^T|)x^*(e) - |\bar{L}_e^T|$ is minimum. We have the following.

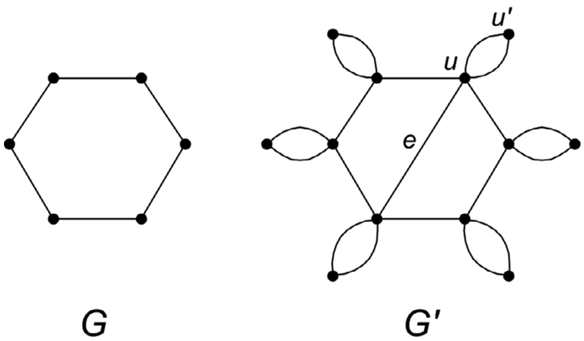

Fig. 5. Transformation for the separation of T-completion inequalities.

Proposition 5. *The separation problem for the T-completion inequalities is NP-hard.*

Proof. The separation problem for the T-completion inequalities consists, given a nonnegative vector $x \in \mathbb{R}^E$, of finding an edge set T, an edge e, and the cardinalities of $L_e^T, \bar{L}_e^T \subseteq E \setminus (T \cup \{e\})$ such that

$$\sum_{f \in E \setminus (T \cup \{e\})} x(f) + \left(|\bar{L}_e^T| - |L_e^T|\right)x(e) - |\bar{L}_e^T| < 0.$$

We use a reduction from the 2-edge connected subgraph problem. We transform the computation of the minimum cardinality 2-edge connected subgraph to the computation of $|\bar{L}_e^T|$.

Let $G = (V, E)$ be a graph, and let $G' = (V', E')$ be the graph obtained from G by adding a copy $u' \in V'$ for every node $u \in V$ and two parallel edges between every node $u \in V$ and its copy $u' \in V'$ as well as an edge e between an arbitrary pair of nodes in V' (see Fig. 5 for illustration).

Determining the minimum cardinality of a 2-edge connected spanning subgraph of G is equivalent to determining the cardinality of $\bar{L}_e^T$. As the former problem is NP-hard, we have that the computation of $|\bar{L}_e^T|$ is also NP-hard, and then the statement follows. □

It follows from the proof of Proposition 5 that the separation of the T-completion inequalities is NP-hard even when T is fixed.

Light T-completion inequalities

As T-completion inequalities are difficult to separate, in what follows, we propose a subclass of these inequalities that are polynomially separable. The light T-completion inequalities are the ones where

- T is a singleton and
- $e \in \delta(V(T))$.

In order to illustrate the light T-completion inequalities, consider the graph G of Fig. 6a together with the tree $T = \{e'\}$ and edge $e \in E \setminus T$. Figure 6b and 6c illustrates L_e^T and $\bar{L}_e^T$, respectively.

Remark 2. If $k = 1$, then $L_e^T = \bar{L}_e^T = \emptyset$ and the associated inequality (23) is redundant.

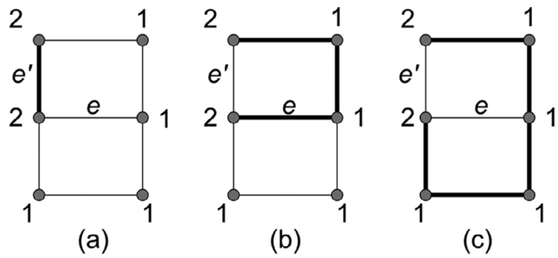

Fig. 6. Graph G, T, L_e^T and $\bar{L}_e^T$.

Lemma 3. *For $e' = uv$, if $\min\{r(u), r(v)\} = 2$, L_e^T and $\bar{L}_e^T$ can be computed in polynomial time.*

Proof. Clearly, computing L_e^T and $\bar{L}_e^T$ is nothing but computing the shortest paths between u and v in $(V, E \setminus \{e\})$ and in $(V, E \setminus \{e, e'\})$, respectively. □

In Fig. 6, $|L_e^T| = 2$ and $|\bar{L}_e^T| = 5$, then the associated inequality (23) looks as

$$\sum_{f \in E \setminus (T \cup \{e\})} x(f) + 3x(e) \geq 5.$$

Note, that the light T-completion inequalities may be not strong, but as it will turn out they are useful to initialize the first mathematical model of the branch-and-cut algorithm.

6. Numerical results

The branch-and-cut algorithm described in the previous section has been implemented in C++, using CPLEX (Cplex, 2009) to manage the branch-and-cut tree and also as an LP solver, and all flow problems are solved using Lemon library (Dezső et al., 2011). CPLEX cuts as well as the preprocessing phase are deactivated. The algorithms have been implemented in C++ and tested on a machine equipped with an Intel(R) Xeon(R) CPU E5-4627 v2 of 3.30 GHz and 504 GB RAM, running under Linux 64 bits. We considered a time-limit of three hours and a maximum of one thread. We use two kinds of instances, the SNDlib (Orlowski et al., 2010) and TSPlib (Reinelt, 1991) instances. In this paper, SNDlib and TSPlib instances are considered in their complete graph-shapes. Nodes are randomly assigned to different connectivity types based on the given parameters. \$CT1 $\times$ nb_nodes\$ nodes assigned to connectivity type1, \$CT2 $\times$ nb_nodes\$ nodes to type 2, and the remaining nodes to type 3. When the specified density "Den" is less than 100%, random links are iteratively removed from the network until the target density is achieved, while keeping the feasibility of the instance. The first mathematical model is initialized with the following cut constraints:

$$x(\delta(u)) \geq \min_{v \in V}(r(v)) \qquad \text{for all } u \in V, \tag{30}$$

together with the light T-completion inequalities. These are generated for every edge $e' \in E$ with $T = \{e'\}$ and every edge $e \in \delta(V(T))$ with $|\bar{L}_e^T| - |L_e^T| \geq 2$.

In the tables below, we have the following entries:

Instance	:	the instance name.
CT1	:	the proportion of nodes with connectivity type 1.
CT2	:	the proportion of nodes with connectivity type 2.
Den	:	the density of the graph.
nb_nodes	:	the number of vertices in V.
nb_Cut	:	the number of generated cut inequalities.
nbSP	:	the number of generated SP-partition inequalities.
nbFP	:	the number of generated F-partition inequalities.
nbP	:	the number of generated partition inequalities.
nbT	:	the number of T-completion inequalities added in the first model.
Tree_Size	:	the number of nodes in the branching tree.
Gap	:	the relative error between the best upper and lower bounds of the branching tree (the gap is zero if the optimal solution is found).
Time	:	the CPU time given by the branch-and-cut algorithm in seconds.
UB	:	the best feasible solution obtained.

Tables 1 and 2 report the results for SNDlib instances, without and with the valid inequalities, respectively, and using the reduction operations as a preprocessing phase before any separation. Each instance is given by a complete graph, where the distance between two nodes is computed using the nodes' coordinates. We consider seven instances for each type and vary the density of the graphs (1.0 or 0.5) by removing randomly some edges. Moreover, we vary the connectivity types of the nodes. Two types of instances are considered: *balanced instances*, where each connectivity type, among 1, 2, and 3, is assigned to 33% of the nodes, and *uniform instances*, where 100% of the nodes have connectivity type 3.

From Table 1, it appears that all the instances could be solved to optimality except one with a 6% of optimality gap. Also, we can remark that the density of the graph has an important impact on the resolution of the problem. Indeed, as it can be noted, when the density increases (equal to 1), the CPU time, the tree size, and the number of generated cuts, all of them increase. We can also remark that the nonuniform (balanced) case is much harder than the uniform one. Instance cost266, for instance, in the uniform case with density 0.5, needed 3.42 seconds, 357 cuts, and 56 generated nodes in the branching tree. while in the balanced case, the outputs were 1367.24 seconds, 19,941, and 23,205, respectively.

From Table 2, it can be seen that, although the same instances as in Table 1 (without adding valid inequalities) have been solved, all the performances have been improved. The maximum CPU time has been decreased by 34%, the maximum number of generated inequalities by 6.3%, and the maximum number of generated nodes in the branching tree by 42.6%. Furthermore, we can remark that all the valid inequalities have been generated. Partition inequalities are the most generated ones reaching 1069 inequalities. On the other hand, the T-completion inequalities have not been much generated. This was predictable since these inequalities are automatically satisfied when the density is 1.

Tables 3 and 4 report the results for the instances from the TSPlib without and with the valid inequalities, respectively, and using the reduction operations. These are larger than those of the SNDlib and seem to be harder to solve.

Table 1
Numerical results on SNDlib instances without valid inequalities

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	nbSP	nbFP	nbP	nb$T$	Tree_Size	Gap	Time	UB
atlanta	15	0.33	0.33	0.5	338	0	0	0	0	37	0.00	0.06	4071
				1	1124	0	0	0	0	252	0.00	0.67	2631
		0	0	0.5	0	0	0	0	0	1	0.00	0.00	4920
				1	474	0	0	0	0	80	0.00	0.21	3265
cost266	37	0.33	0.33	0.5	10870	0	0	0	0	3849	0.00	111.84	275
				1	19941	0	0	0	0	23205	0.00	1 367.24	199
		0	0	0.5	51	0	0	0	0	4	0.00	0.10	380
				1	357	0	0	0	0	56	0.00	3.42	275
france	25	0.33	0.33	0.5	1263	0	0	0	0	73	0.00	0.57	3841
				1	3876	0	0	0	0	648	0.00	9.32	2379
		0	0	0.5	51	0	0	0	0	15	0.00	0.09	5119
				1	1020	0	0	0	0	92	0.00	1.28	3254
india	35	0.33	0.33	0.5	7903	0	0	0	0	1650	0.00	43.73	451
				1	8145	0	0	0	0	6228	0.00	270.41	347
		0	0	0.5	51	0	0	0	0	7	0.00	0.14	652
				1	408	0	0	0	0	64	0.00	3.56	452
india30	30	0.33	0.33	0.5	799	0	0	0	0	34	0.00	0.50	449
				1	1892	0	0	0	0	259	0.00	6.79	326
		0	0	0.5	102	0	0	0	0	5	0.00	0.07	568
				1	153	0	0	0	0	8	0.00	0.21	418
norway20	20	0.33	0.33	0.5	2397	0	0	0	0	283	0.00	1.13	5885
				1	2106	0	0	0	0	469	0.00	3.18	3759
		0	0	0.5	153	0	0	0	0	30	0.00	0.09	7839
				1	612	0	0	0	0	88	0.00	0.56	4814
ta2	65	0.33	0.33	0.5	26802	0	0	0	0	31127	0.00	10 066.20	5539
				1	24690	0	0	0	0	22964	0.06	–	4133
		0	0	0.5	51	0	0	0	0	5	0.00	1.23	7587
				1	867	0	0	0	0	217	0.00	169.53	5334

In Table 3, which displays the results without the use of valid inequalities, only 25% of the instances could be solved within the time limit, and the optimality gap reached 28%. Also, the instances that could be solved are among the uniform ones with a density of 0.5. However, it clearly appears from Table 4, in which the valid inequalities are used, that, as for the SNDlib instances, these inequalities have been effective. The CPU time has been decreased for all instances solved to optimality. For instance, for bier127 and ch130, the decrease is 87% and 70%, respectively. Also, instance eil101 with density 1, in the uniform case, was solved to optimality while without adding valid inequalities it could not be solved after three hours. Finally, it can be noted that there is a significant improvement in the number of generated valid inequalities and nodes in the branching tree. However, we observe that no T-completion inequalities have been generated. This may be explained, on the one hand, as before, by the fact that these inequalities are unlikely to be met when the density is relatively high. Preliminary experiments, with a small density (around 10%), have shown, on the contrary, that T-completion inequalities may be generated in important numbers. On the other hand, TSPlib instances have big sizes (more than 96 nodes) and this also

Table 2
Numerical results on SNDlib instances with valid inequalities

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	nbSP	nbFP	nbP	nb$T$	Tree_Size	Gap	Time	UB
atlanta	15	0.33	0.33	0.5	811	0	4	0	9	53	0.00	0.11	4071
				1	1186	3	14	0	0	295	0.00	0.83	2631
		0	0	0.5	0	0	2	0	22	1	0.00	0.00	4920
				1	285	15	25	20	0	39	0	0.14	3265
cost266	37	0.33	0.33	0.5	8751	0	13	426	0	2569	0.00	82.36	275
				1	11001	5	19	533	0	4320	0.00	339.65	199
		0	0	0.5	51	0	1	2	1	3	0.00	0.11	380
				1	357	9	4	5	0	13	0	1.71	275
france	25	0.33	0.33	0.5	1324	0	3	16	2	75	0.00	0.56	3841
				1	4247	0	10	70	0	695	0.00	11.41	2379
		0	0	0.5	0	1	8	1	4	9	0.00	0.08	5119
				1	306	14	8	9	0	25	0	0.51	3254
india	35	0.33	0.33	0.5	6029	1	7	95	0	755	0.00	20.01	451
				1	5362	0	13	185	0	1895	0.00	90.32	347
		0	0	0.5	0	5	12	7	0	18	0.00	0.62	652
				1	102	4	9	4	0	13	0	1.12	452
india30	30	0.33	0.33	0.5	785	0	0	7	1	27	0.00	0.39	449
				1	1271	0	0	13	0	189	0.00	5.20	326
		0	0	0.5	51	0	0	1	3	4	0.00	0.07	568
				1	204	5	8	2	0	6	0	0.48	418
norway20	20	0.33	0.33	0.5	2256	2	13	32	3	199	0.00	0.90	5885
				1	1354	1	6	37	0	212	0.00	1.38	3759
		0	0	0.5	51	4	10	10	12	28	0.00	0.10	7839
				1	102	9	2	4	0	8	0	0.09	4814
ta2	65	0.33	0.33	0.5	25193	1	51	1069	0	17866	0.00	6 549.13	5539
				1	22469	4	49	630	0	13882	0.03	–	4078
		0	0	0.5	102	1	2	1	0	4	0.00	1.32	7587
				1	561	16	40	13	0	68	0	102.49	5334

leaves T-completion to be unlikely violated, that is to say, to find edges $e' \in E$ and $e \in \delta(V(T))$ with $|\bar{L}_e^T| - |L_e^T| \geq 2$, where $T = \{e'\}$.

Table 5 shows a comparison of the results between our branch-and-cut algorithm using our cuts only and the same algorithm using Cplex cuts only. The results are obtained for the SNDlib instances. Notably, Cplex cuts significantly enhance the number of nodes in the branching tree, leading to positive impacts on the CPU time. Meanwhile, our branch-and-cut algorithm, using only our cuts, shows improvements in both CPU time and the number of branching nodes for two uniform case instances ($CT1 = CT2 = 0$): Atlanta with a density of 50% and France with a density of 100%.

Table 6 provides a comparison of the results for the TSPlib instances. Our branch-and-cut algorithm, using only our cuts, achieves better CPU time performance on three out of seven instances solved within the time limit. These instances are uniform ($CT1 = CT2 = 0$) with 50% of density: bier127, eil101, and gr96. Both algorithms reach the time limit on the same instances.

Table 3
Numerical results on TSPlib instances without valid inequalities

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	nbSP	nbFP	nbP	nb$T$	Tree_Size	Gap	Time	UB
bier127	127	0.33	0.33	0.5	23711	0	0	0	0	2206	0.08	–	175003
				1	15192	0	0	0	0	750	0.28	–	161677
		0	0	0.5	816	0	0	0	0	115	0.00	3 554.22	226838
				1	4437	0	0	0	0	460	0.20	–	187735
ch130	130	0.33	0.33	0.5	22107	0	0	0	0	1741	0.12	–	10084
				1	8953	0	0	0	0	516	0.23	–	8245
		0	0	0.5	459	0	0	0	0	97	0.00	2 958.86	12695
				1	2499	0	0	0	0	278	0.15	–	10248
eil101	101	0.33	0.33	0.5	25879	0	0	0	0	8412	0.06	–	1074
				1	25236	0	0	0	0	2723	0.15	–	836
		0	0	0.5	153	0	0	0	0	6	0.00	9.37	1478
				1	3519	0	0	0	0	1316	0.02	–	1038
gr120	120	0.33	0.33	0.5	26619	0	0	0	0	2959	0.07	–	3183
				1	13592	0	0	0	0	1141	0.12	–	2326
		0	0	0.5	306	0	0	0	0	23	0.00	77.62	4076
				1	2652	0	0	0	0	484	0.05	–	2882
gr96	96	0.33	0.33	0.5	23811	0	0	0	0	9694	0.04	–	1043
				1	19509	0	0	0	0	3620	0.11	–	758
		0	0	0.5	408	0	0	0	0	17	0.00	27.90	1412
				1	3672	0	0	0	0	1911	0.03	–	940
kroA100	100	0.33	0.33	0.5	35180	0	0	0	0	6553	0.07	–	36044
				1	19971	0	0	0	0	2868	0.20	–	32351
		0	0	0.5	1275	0	0	0	0	285	0.00	936.52	45748
				1	3672	0	0	0	0	909	0.07	–	34839
pr124	120	0.33	0.33	0.5	19606	0	0	0	0	2898	0.12	–	129472
				1	9968	0	0	0	0	1122	0.26	–	114252
		0	0	0.5	867	0	0	0	0	115	0.00	4 687.51	160081
				1	1938	0	0	0	0	324	0.18	–	125758

Additionally, our branch-and-cut algorithm improves the upper bound on nine instances compared to the version using only Cplex cuts. It's worth noting that the branch-and-cut algorithm with only Cplex cuts faces more difficulty with TSPlib instances than with SNDlib instances. Larger instances appear to be more challenging for Cplex. Overall, the branch-and-cut algorithm with Cplex cuts remains more efficient, mainly due to the separation algorithms that consume significant computational time.

Table 7 reports numerical results without using the reduction operations. The aim is to show the impact of the operations on the resolution of the problem. To this end, we just consider some SNDlib instances where all nodes have connectivity type 3, the density is equal to 1, and all valid cuts are enabled. It appears that the preprocessing procedures considerably improved the performance of the algorithm. For instance, for ta2, and cost266 the improvement in the CPU time is 17%, and 74.4%, respectively. In addition, the preprocessing improves the number of generated cuts and the number of nodes in the branching tree. In norway20, for instance, the number of branching nodes decreased from 28 to 8 nodes, and in France, from 49 to 25.

Table 4
Numerical results on TSPlib instances with valid inequalities

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	nbSP	nbFP	nbP	nb$T$	Tree_Size	Gap	Time	UB
bier127	127	0.33	0.33	0.5	22778	2	5	33	0	1911	0.11	–	180532
				1	14922	8	0	8	0	848	0.26	–	158677
		0	0	0.5	663	4	9	9	0	22	0.00	460.47	226838
				1	2805	19	6	4	0	104	0.23	–	196148
ch130	130	0.33	0.33	0.5	19597	5	5	51	0	1611	0.09	–	9829
				1	7539	6	2	7	0	414	0.26	–	8554
		0	0	0.5	459	4	12	4	0	25	0.00	879.60	12695
				1	1887	23	0	5	0	90	0.19	–	10763
eil101	101	0.33	0.33	0.5	32287	7	18	273	0	7374	0.07	–	1083
				1	20439	13	4	95	0	2287	0.13	–	819
		0	0	0.5	255	6	9	2	0	9	0.00	34.96	1478
				1	1938	55	46	47	0	311	0.00	7 411.67	1029
gr120	120	0.33	0.33	0.5	24671	2	5	153	0	2462	0.07	–	3190
				1	12656	1	1	21	0	965	0.12	–	2327
		0	0	0.5	153	5	8	2	0	13	0.00	152.73	4076
				1	2550	11	12	9	0	527	0.05	–	2882
gr96	96	0.33	0.33	0.5	27955	7	10	320	0	7791	0.04	–	1039
				1	18502	10	17	97	0	2952	0.11	–	763
		0	0	0.5	153	3	1	3	0	9	0.00	21.06	1412
				1	2397	73	62	66	0	540	0.01	–	938
kroA100	100	0.33	0.33	0.5	31880	16	27	210	0	5849	0.12	–	37446
				1	18151	9	3	89	0	2155	0.14	–	29969
		0	0	0.5	612	15	28	19	0	99	0.00	549.51	45748
				1	3468	60	14	28	0	566	0.07	–	34724
pr124	120	0.33	0.33	0.5	18755	9	0	16	0	2007	0.09	–	124650
				1	9885	9	1	2	0	780	0.26	–	114252
		0	0	0.5	765	10	18	17	0	70	0.00	2 921.43	160081
				1	1428	18	0	1	0	134	0.29	–	144936

7. Concluding remarks

In this paper, we have considered the survivable network design problem in the nonuniform case where the connectivity types are in {0, 1, 2, 3}. To the best of our knowledge, this is the first study of the problem in this case. We have developed a cutting-plane-based approach for the problem, based on a deep investigation of the associated polytope. We have described several classes of valid inequalities. In particular, we have introduced a new class called T-completion inequalities induced by trees of the underlying graph. We have given necessary conditions and sufficient conditions for the valid inequalities to be facet defining and devise separation routines. These take into account the conditions for the inequalities to define facets. We have also shown that the separation problem for the new inequalities is NP-hard. Using these theoretical results, we have proposed a branch-and-cut algorithm for the problem. The experimental results have shown the effectiveness of the valid inequalities. In fact, most of the big-sized instances that could not be solved without valid inequalities have been solved to optimality when these have been added. It has also appeared that the reduction operations improve considerably the efficiency of the algorithm.

Table 5
Numerical results on SNDlib instances with CPLEX' cuts

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	Tree_Size	Gap	Time	UB
atlanta	15	0.33	0.33	0.5	753	19	–	0.10	4071
				1	1379	86	–	0.41	2631
		0	0	0.5	153	1	–	0.03	4920
				1	44	1	–	0.01	3265
cost266	37	0.33	0.33	0.5	8313	621	–	22.87	275
				1	10437	2479	–	173.52	199
		0	0	0.5	51	1	–	0.04	380
				1	459	8	–	1.10	275
france	25	0.33	0.33	0.5	1807	53	–	0.56	3841
				1	1866	128	–	2.51	2379
		0	0	0.5	0	1	–	0.01	5119
				1	867	61	–	1.13	3254
india	35	0.33	0.33	0.5	4082	334	–	10.32	451
				1	4884	1329	–	73.57	347
		0	0	0.5	51	8	–	0.24	652
				1	153	7	–	0.47	452
india30	30	0.33	0.33	0.5	590	1	–	0.13	449
				1	1487	13	–	0.96	326
		0	0	0.5	51	1	–	0.04	568
				1	153	5	–	0.17	418
norway20	20	0.33	0.33	0.5	1996	75	–	0.48	5885
				1	1274	17	–	0.33	3759
		0	0	0.5	51	1	–	0.03	7839
				1	306	1	–	0.08	4814
ta2	65	0.33	0.33	0.5	13883	1694	–	687.71	5539
				1	20391	12843	0.01	–	4071
		0	0	0.5	51	1	–	0.62	7587
				1	765	103	–	99.77	5334

A comparison of the branch-and-cut algorithm with and without the Cplex cuts shows that the one using Cplex cuts performs better for most of the instances, in particular for the medium-sized ones as those of the SNDlib. It is significantly much faster. However, it appeared that our algorithm may perform better for big-sized instances. This can be explained by the fact that the separation routines consume a lot of time. One of our future research lines is to study more deeply the separation routines and improve their performances.

Moreover, it would be interesting to characterize cases in which the new valid inequalities may define facets. Also, it would be of great interest to study the problem with the additional constraint that the length of the paths should not exceed a given threshold L. Here the length of the paths is given in terms of the number of edges of the paths between the pairs of nodes. This variant is known as the hop-constrained survivable network design problem (Kerivin and Mahjoub, 2005a; Huygens et al., 2007). The QoS-bounded paths constraints permit effective routing in telecommunication networks in the event of failure of a link (or more) in the network.

Table 6
Numerical results on TSPlib instances with CPLEX' cuts

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	Tree_Size	Gap	Time	UB
bier127	127	0.33	0.33	0.5	9917	198	0.17	–	196145
				1	2346	84	0.17	–	180566
		0	0	0.5	969	79	0.00	22599	226838
				1	2244	54	0.17	–	182631
ch130	130	0.33	0.33	0.5	16499	578	0.07	–	9886
				1	8015	176	0.33	–	9878
		0	0	0.5	561	35	0.00	1652.50	12695
				1	1734	52	0.20	–	11034
eil101	101	0.33	0.33	0.5	26110	3080	0.02	–	1068
				1	14828	972	0.09	–	826
		0	0	0.5	153	5	–	17.76	1478
				1	2550	595	0.01	–	1035
gr120	120	0.33	0.33	0.5	18929	1447	0.03	–	3134
				1	10304	531	0.10	–	2339
		0	0	0.5	204	13	–	74.50	4076
				1	2397	174	0.04	–	2889
gr96	96	0.33	0.33	0.5	19062	3663	0.01	–	1035
				1	16544	1858	0.07	–	753
		0	0	0.5	255	14	–	52.85	1412
				1	3264	1327	0.01	–	938
kroA100	100	0.33	0.33	0.5	20706	3085	0.04	–	35590
				1	14439	1599	0.14	–	30636
		0	0	0.5	663	89	0.00	287.23	45748
				1	3519	550	0.12	–	36940
pr124	120	0.33	0.33	0.5	18158	1629	0.13	–	133373
				1	10121	522	0.26	–	115473
		0	0	0.5	816	77	0.00	3126.95	160081
				1	2244	382	0.17	–	125119

Table 7
Numerical results on SNDlib instances without the reduction operations

Instance	nb_nodes	CT1	CT2	Den	nb_Cut	nbSP	nbFP	nbP	nb$T$	Tree_Size	Gap	Time	UB
atlanta	15	0	0	1	393	19	38	17	0	47	0	0.20	3265
cost266	37	0	0	1	306	5	9	15	0	29	0	1.92	275
france	25	0	0	1	663	20	15	16	0	49	0	0.99	3254
india	35	0	0	1	306	7	27	9	0	53	0	4.38	452
india30	30	0	0	1	14	2	2	1	0	4	0	0.44	418
norway20	20	0	0	1	306	10	10	10	0	28	0	0.28	4814
ta2	65	0	0	1	561	14	33	13	0	70	0	124.02	5334

Acknowledgements

We would like to thank the anonymous referees for their valuable comments that permitted to significantly improve the presentation of the paper.

References

Ameln, M., Sand Fuglum, J., Thun, K., Andersson, H., Stålhane, M., 2021. A new formulation for the liner shipping network design problem. *International Transactions in Operational Research* 28, 2, 638–659.

Babarczi, P., Tapolcai, J., 2012. Protection survivability architectures: principles and challenging issues. In *Resilient Optical Network Design: Advances in Fault-Tolerant Methodologies*. IGI Global, Hershey, PA, pp. 27–56.

Baïou, M., Barahona, F., Mahjoub, A.R., 2000. Separation of partition inequalities. *Mathematics of Operations Research* 25, 2, 243–254.

Baïou, M., Mahjoub, A.R., 1997. Steiner 2-edge connected subgraph polytopes on series-parallel graphs. *SIAM Journal on Discrete Mathematics* 10, 3, 505–514.

Barahona, F., 1992. Separating from the dominant of the spanning tree polytope. *Operations Research Letters* 12, 4, 201–203.

Bendali, F., Diarrassouba, I., Mahjoub, A.R., Didi Biha, M., Mailfert, J., 2010. A branch-and-cut algorithm for the k-edge connected subgraph problem. *Networks* 55, 1, 13–32.

Biha, M.D., Kerivin, H., Mahjoub, A.R., 2008. On the polytope of the (1, 2)-survivable network design problem. *SIAM Journal on Discrete Mathematics* 22, 4, 1640–1666.

Biha, M.D., Mahjoub, A.R., 1996. k-Edge connected polyhedra on series-parallel graphs. *Operations Research Letters* 19, 2, 71–78.

Biha, M.D., Mahjoub, A.R., 2000. Steiner k-edge connected subgraph polyhedra. *Journal of Combinatorial Optimization* 4, 131–144.

Blanco, V., Puerto, J., 2022. On hub location problems in geographically flexible networks. *International Transactions in Operational Research* 29, 4, 2226–2249.

Chen, H., Wang, X., Liu, Z., Li, Z., Shen, L., 2022. Survivable networks for consensus. *IEEE Transactions on Control of Network Systems* 9, 2, 588–600.

Chopra, S., 1994. The k-edge-connected spanning subgraph polyhedron. *SIAM Journal on Discrete Mathematics* 7, 2, 245–259.

Cplex,2009. V12.6: User's manual for Cplex. *International Business Machines Corporation* 46, 53, 157.

Cunningham, W.H., 1985. Optimal attack and reinforcement of a network. *Journal of the ACM* 32, 3, 549–561.

Dezső, B., Jüttner, A., Kovács, P., 2011. Lemon–an open source C++ graph template library. *Electronic Notes in Theoretical Computer Science* 264, 5, 23–45.

Diarrassouba, I., Kütücü, H., Mahjoub, A.R., 2016. Two node-disjoint hop-constrained survivable network design and polyhedra. *Networks* 67, 4, 316–337.

Diarrassouba, I., Mahjoub, M., Mahjoub, A.R., Taktak, R., 2017. The survivable k-node-connected network design problem: valid inequalities and branch-and-cut. *Computers & Industrial Engineering* 112, 690–705.

Duhamel, C., Santos, A.C., 2024. The strong network orientation problem. *International Transactions in Operational Research* 31, 1, 192–220.

Eshoul, A., Mouftah, H.T., 2012. Performance evaluation of survivability approaches in optical networks. In *Resilient Optical Network Design: Advances in Fault-Tolerant Methodologies*. IGI Global, Hershey, PA, pp. 134–157.

Fleischer, L., 2001. A 2-approximation for minimum cost {0, 1, 2} vertex connectivity. In *LNCS*, Vol. 2081, Springer, Berlin, pp. 115–129.

Fleischer, L., Jain, K., Williamson, D.P., 2006. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences* 72, 5, 838–867.

Fouilhoux, P., Karasan, O.E., Mahjoub, A.R., Özkök, O., Yaman, H., 2012. Survivability in hierarchical telecommunications networks. *Networks* 59, 1, 37–58.

Gajić, M., Lange, S., Vatten, T., Furdek, M., Heegaard, P.E., 2023. Survivability assessment of 5G network slicing during massive outages. In *2023 13th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, Piscataway, NJ, pp. 1–8.

Garey, M.R., Johnson, D.S., 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, CA.

Goemans, M.X., Bertsimas, D.J., 1993. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming* 60, 1-3, 145–166.

Gomory, R.E., Hu, T.C., 1961. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics* 9, 4, 551–570.

Gouveia, L., Leitner, M., 2017. Design of survivable networks with vulnerability constraints. *European Journal of Operational Research* 258, 1, 89–103.

Grötschel, M., Monma, C.L., Stoer, M., 1992a. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research* 40, 2, 309–330.

Grötschel, M., Monma, C.L., Stoer, M., 1992b. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM Journal on Optimization* 2, 3, 474–504.

Grötschel, M., Monma, C.L., Stoer, M., 1995. Design of survivable networks. *Handbooks in Operations Research and Management Science* 7, 617–672.

Gusfield, D., 1990. Very simple methods for all pairs network flow analysis. *SIAM Journal of Computing* 19, 1, 143–155.

Hochbaum, D.S., 1997. Approximation algorithms for NP-hard problems. *ACM SIGACT News* 28, 2, 40–52.

Huygens, D., Labbé, M., Mahjoub, A.R., Pesneau, P., 2007. The two-edge connected hop-constrained network design problem: valid inequalities and branch-and-cut. *Networks* 49, 1, 116–133.

Jain, K., 2001. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica* 21, 39–60.

Kerivin, H., 2000. Réseaux fiables et polyèdres. Ph.D. thesis, Clermont-Ferrand 2.

Kerivin, H., Mahjoub, A.R., 2002. Separation of partition inequalities for the (1, 2)-survivable network design problem. *Operations Research Letters* 30, 4, 265–268.

Kerivin, H., Mahjoub, A.R., 2005a. Design of survivable networks: a survey. *Networks* 46, 1, 1–21.

Kerivin, H., Mahjoub, A.R., 2005b. On survivable network polyhedra. *Discrete Mathematics* 290, 2-3, 183–210.

Kerivin, H., Mahjoub, A.R., Nocq, C., 2004. (1, 2)-Survivable networks: facets and branch-and-cut. In *The Sharpest Cut: The Impact of Manfred Padberg and his work*. SIAM, Philadelphia, PA, pp. 121–152.

Larsson, C., 2014. *Design of Modern Communication Networks: Methods and Applications*. Academic Press, New York.

Leu, G., Tang, J., 2019. Survivable networks via UAV swarms guided by decentralized real-time evolutionary computation. In *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Piscataway, NJ, pp. 1945–1952.

Mahjoub, A.R., 1994. Two-edge connected spanning subgraphs and polyhedra. *Mathematical Programming* 64, 1-3, 199–208.

Mahjoub, A.R., Nocq, C., 1999. On the linear relaxation of the 2-node connected subgraph polytope. *Discrete Applied Mathematics* 95, 1-3, 389–416.

Mahjoub, A.R., Pesneau, P., 2008. On the Steiner 2-edge connected subgraph polytope. *RAIRO-Operations Research* 42, 3, 259–283.

Mahjoub, M., 2017. The survivable network design problems with high node-connectivity constraints: polyhedra and algorithms. Ph.D. thesis, Université Paris-Dauphine; Université de Tunis El Manar.

Menger, K., 1927. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae* 10, 1, 96–115.

Monma, C.L., Shallcross, D.F., 1989. Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research* 37, 4, 531–541.

Olivera, A., Amoza, F.R., Testuri, C.E., 2010. A grasp algorithm for a capacitated, fixed charge, multicommodity network flow problem with uncertain demand and survivability constraints. *International Transactions in Operational Research* 17, 6, 765–776.

Orlowski, S., Wessäly, R., Pióro, M., Tomaszewski, A., 2010. Sndlib 1.0'survivable network design library. *Networks* 55, 3, 276–286.

Patricio, P.F., 2024. Wildlife reserve design with connectivity and buffer requirements. *International Transactions in Operational Research* 31, 4, 2231–2246.

Raghavan, S., 2004. Low-connectivity network design on series-parallel graphs. *Networks* 43, 3, 163–176.

Ravi, R., Williamson, D.P., 1997. An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica* 18, 21–43.

Reinelt, G., 1991. TSPLIB–a traveling salesman problem library. *ORSA Journal on Computing* 3, 4, 376–384.

Rezk, N., Shehata, M., Gasser, S.M., El-Mahallawy, M.S., 2023. Survivable network slicing using shared link protection. *IEEE Access* 11, 81900–81913.

Slama, L., 2008. Conception de réseaux fiables: séparation et polyèdres. Ph.D. thesis, Clermont-Ferrand 2, Clermont-Ferrand, France.

Steiglitz, K., Weiner, P., Kleitman, D., 1969. The design of minimum-cost survivable networks. *IEEE Transactions on Circuit Theory* 16, 4, 455–460.

Stoer, M., 1992. *Design of Survivable Networks*. Lecture Notes in Mathematics, Vol. 1531. Springer, Berlin.

Takamizawa, K., Nishizeki, T., Saito, N., 1982. Linear-time computability of combinatorial problems on series-parallel graphs. *Journal of the ACM* 29, 3, 623–641.

Magnanti, T.L., Raghavan, S., 1999. A dual-ascent algorithm for low-connectivity network design. Technical report, OR-Center, Massachusetts Institute of Technology, Cambridge, MA.

Wang, B., 2012. Survivability in optical networks: principles and state-of-the-art. In *Resilient Optical Network Design: Advances in Fault-Tolerant Methodologies*. IGI Global, Hershey, PA, pp. 1–26.

Wierzbicki, A.P., Burakowski, W., 2011. A conceptual framework for multiple-criteria routing in qos ip networks. *International Transactions in Operational Research* 18, 3, 377–399.

Winter, P., 1985. Generalized Steiner problem in Halin graphs. Paper presented at Proceedings of the 12th International Symposium on Mathematical Programming, Boston, MA.

Winter, P., 1986. Generalized Steiner problem in series-parallel networks. *Journal of Algorithms* 7, 4, 549–566.

Winter, P., 1987a. Steiner problem in Halin networks. *Discrete Applied Mathematics* 17, 3, 281–294.

Winter, P., 1987b. Steiner problem in networks: a survey. *Networks* 17, 2, 129–167.

Appendix A

A.1. Proof of Theorem 2

1. Suppose, for instance, that $G(V_1)$ is not connected. Let V_1^1 and V_1^2 be a partition of V_1 such that $[V_1^1, V_1^2] = \emptyset$ and $V_1^1 \neq \emptyset \neq V_1^2$.
   Let $\pi' = (V_1^1, V_1^2, V_2, \ldots, V_p)$. We have that

$$x(\delta(\pi')) = x(\delta(\pi)). \tag{A1}$$

   However, the right-hand side of inequality (7) with respect to π' is greater than that related to π. Hence, (7) is dominated by the SP-partition induced by π'. Hence, it cannot define a facet.
2. Suppose that G_π is not 2-node connected. Let $G_\pi^1 = (W_1, \Gamma_1)$ and $G_\pi^2 = (W_2, \Gamma_2)$ be the two pieces of G_π such that $\Gamma_1 \cap \Gamma_2 = \emptyset$ and $W_1 \cap W_2 = \{w\}$ for some node w of G_π.
   Let $W_1 = \{w_1 = w, w_2, \ldots, w_{p_1(\pi)}\}$, $W_2 = \{w_{p_1(\pi)+1} = w, w_{p_1(\pi)+2}, \ldots, w_{p_1(\pi)+p_2(\pi)}\}$.
   Note that each node w_i corresponds to a set V_i of partition π. Also note that $p = p_1(\pi) + p_2(\pi) - 1$. Let G^1 and G^2 be the graphs obtained by expanding the nodes of G_π^1 and G_π^2, respectively. Suppose, without loss of generality, that w_i (resp. w_j) corresponds to V_i (resp. V_{j-1}) for $i = 1, \ldots, p_1(\pi)$ (resp. $j = p_1(\pi) + 2, \ldots, p_1(\pi) + p_2(\pi)$).
   Let $\pi_1 = (V_1 \cup \bigcup_{i=p_1(\pi)+1}^{p} V_i, V_2, \ldots, V_p)$, $\pi_2 = (\bigcup_{i=1}^{p_1(\pi)} V_i, V_{p_1(\pi)} + 1, \ldots, V_{p_1(\pi)+p_2(\pi)})$. Note that π_1 and π_2 are partitions of V. Suppose $r(\pi_1) = r(\pi_2) = 3$. Hence by Lemma 1, $r(\pi) = 3$. Moreover, we have

$$x(\delta(\pi_1)) \geq p_1(\pi) + p_3(\pi_1) - 1, \tag{A2}$$
$$x(\delta(\pi_2)) \geq p_2(\pi) + p_3(\pi_2) - 1. \tag{A3}$$

By summing (A2) and (A3), we get

$$
\begin{aligned}
x(\delta(\pi)) &= x(\delta(\pi_1)) + x(\delta(\pi_2)) && \geq p_1(\pi) + p_2(\pi) - 1 + p_3(\pi_1) + p_3(\pi_2) - 1 \\
& && = p + p_3(\pi_1) + p_3(\pi_2) - 1 \\
& && \geq p + p_3(\pi) - 1.
\end{aligned}
$$

The last inequality comes from the fact that $p_3(\pi_1) + p_3(\pi_2) \geq p_3(\pi)$. Hence the SP-partition inequality, induced by π, is redundant with respect to those induced by π_1 and π_2. Therefore, it cannot define a facet.

The other cases, depending on the values of $r(\pi_1)$ and $r(\pi_2)$, can be treated along the same line.

3. First of all we may suppose that G_π is 2-node connected, for otherwise Condition (2) applies. Suppose that $G_\pi \setminus e$ is not 2-node connected. Also suppose, on the contrary, that (7) is facet defining. Since e is not essential and (7) is different from the trivial inequality $x(e) \leq 1$, there must exist a solution S of ESNDP(G), not containing e, whose incidence vector satisfies (7) with equality.

   Let

$$
\pi_1 = (V_1 \cup \bigcup_{i=q+1}^{p} V_i, V_2, \dots, V_q),
$$
$$
\pi_2 = (V_1 \cup \bigcup_{i=2}^{q} V_i, V_{q+1}, \dots, V_p).
$$

   We will distinguish three cases.

   Case 1: $r(\pi_1) = r(\pi_2) = 3$. By Lemma 1, it follows that $r(\pi) = 3$. Moreover, we have

$$
\begin{aligned}
& x^S(\delta(\pi_1)) \geq q + p_3(\pi_1) - 1, \\
& x^S(\delta(\pi_2)) \geq p - q + 1 + p_3(\pi_2) - 1,
\end{aligned}
$$

   where x^S denotes the incidence vector of solution S introduced above.

   By summing these inequalities, we obtain that $x^S(\delta(\pi)) = x^S(\delta(\pi_1)) + x^S(\delta(\pi_2)) \geq p + p_3(\pi_1) + p_3(\pi_2) - 1$. If $con(V_1) = 3$, then $p_3(\pi_1) + p_3(\pi_2) = p_3(\pi) + 1$. Consequently, $x^S(\delta(\pi)) \geq p + p_3(\pi)$, and hence inequality (7) is not tight for x^S, a contradiction. If $con(V_1) \neq 3$, as $r(\pi_1) = r(\pi_2) = 3$, there must exist a set V_i, $i \in \{2, \dots, q\}$ (resp. V_j, $j \in \{q+1, \dots, p\}$) with $con(V_i) = 3$ (resp. $con(V_j) = 3$). Since every path between two sets V_i and V_j with $i \in \{2, \dots, q\}$, and $j \in \{q+1, \dots, p\}$ goes through V_1, we can suppose that $con(V_1) = 3$. Hence the previous case applies.

   Case 2: $r(\pi_1) = 3$ and $r(\pi_2) = 2$. The case $r(\pi_1) = 2$, $r(\pi_2) = 3$ is similar. Hence by Lemma 1, $r(\pi) = 3$. Also, we have

$$
\begin{aligned}
& x^S(\delta(\pi_1)) \geq q + p_3(\pi_1) - 1, \\
& x^S(\delta(\pi_2)) \geq p - q + 1.
\end{aligned}
$$

   By summing these inequalities, we obtain that

$$
x^S(\delta(\pi)) = x^S(\delta(\pi_1)) + x^S(\delta(\pi_2)) \geq p + p_3(\pi_1).
$$

Observe here that $p_3(\pi) = p_3(\pi_1)$. Hence

$$\begin{aligned} x^S(\delta(\pi)) &\geq p + p_3(\pi), \\ &> p + p_3(\pi) - 1, \end{aligned}$$

Therefore, inequality (7) is not tight for x^S, a contradiction.
Case 3: $r(\pi_1) = r(\pi_2) = 2$. Then by Lemma 1, $r(\pi) = 2$. Moreover, we have

$$\begin{aligned} &x^S(\delta(\pi_1)) \geq q, \\ &x^S(\delta(\pi_2)) \geq p - q + 1. \end{aligned}$$

This yields

$$\begin{aligned} x^S(\delta(\pi)) &= x^S(\delta(\pi_1)) + x^S(\delta(\pi_2)) \\ &\geq p + 1 \\ &> p, \end{aligned}$$

contradicting the fact that (7) is tight for x^S.
In consequence, one should have either $r(\pi_1) = 1$ or $r(\pi_2) = 2$ or both.

4. Suppose $[V_i^1, V_i^2] \setminus \{e_0\} = \emptyset$. We will distinguish different cases.
Case 1: $con(V_i) = 3$.
Let us assume, on the contrary, that (7) defines (a nontrivial) facet of ESNDP(G). Then there should exist a solution S noncontaining e_0 such that x^S satisfies (7) with equality. Let $\pi' = (V_1, \ldots, V_{i-1}, V_i^1, V_i^2, V_{i+1}, \ldots, V_p)$.
As $con(V_i) = 3$, we have $r(\pi) = r(\pi') = 3$, and in consequence,

$$\begin{aligned} x^S(\delta(\pi)) &= x^S(\delta(\pi')) - x^S([V_i^1, V_i^2]) \\ &\geq p + 1 + p_3(\pi') - 1 - x^S([V_i^1, V_i^2]) \\ &= p + p_3(\pi'). \end{aligned}$$

The last equation comes from the assumption that $x^S([V_i^1, V_i^2]) = 0$ as $e_0 \notin S$. As $p_3(\pi') \geq p_3(\pi)$, it follows that

$$\begin{aligned} x^S(\delta(\pi)) &\geq p + p_3(\pi)) \\ &> p + p_3(\pi)) - 1. \end{aligned}$$

Hence, inequality (7) is not satisfied with equality for x^S, a contradiction.

Case 2: $con(V_i) = 2$ and $p_3(\pi) \neq 0$. Hence, $r(\pi) = r(\pi') = 3$ and $p_3(\pi) = p_3(\pi')$. We have

$$\begin{aligned} x^S(\delta(\pi)) &= x^S(\delta(\pi')) - x^S([V_i^1, V_i^2]) \\ &\geq p + 1 + p_3(\pi') - 1 \\ &= p + p_3(\pi') \\ &= p + p_3(\pi) \\ &> p + p_3(\pi) - 1, \end{aligned}$$

implying that (7) is not tight for x^S, a contradiction.

Case 3: $con(V_i) = 2$ and $p_3(\pi) = 0$. Hence we have $r(\pi) = 2$ and $r(\pi') \geq r(\pi)$. Suppose first that $r(\pi') = r(\pi) = 2$. Therefore,

$$\begin{aligned} x^S(\delta(\pi)) &= x^S(\delta(\pi')) - x^S([V_i^1, V_i^2]) \\ &\geq p + 1 \\ &> p. \end{aligned}$$

Hence inequality (7) is not satisfied with equality for x^S, a contradiction.
If $r(\pi') > r(\pi) = 2$. Then $r(V_i^1) = r(V_i^2) = 3$. We have

$$\begin{aligned} x^S(\delta(\pi)) &= x^S(\delta(\pi')) - x^S([V_i^1, V_i^2]) \\ &\geq p + 1 + p_3(\pi') - 1 \\ &= p + 1, \end{aligned}$$

as $p_3(\pi') = 1$. But this implies that $x^S(\delta(\pi)) > p - 1$, a contradiction.

Case 4: $p_3(\pi) = p_2(\pi) = 0$ and $p_1(\pi) \neq 0$. Then $r(\pi') \geq r(\pi) = 1$. Suppose $r(\pi') = r(\pi) = 1$ (the other case is similar). Then,

$$\begin{aligned} x^S(\delta(\pi)) &= x^S(\delta(\pi')) - x^S([V_i^1, V_i^2]) \\ &\geq p \\ &> p - 1. \end{aligned}$$

In consequence, inequality (7) is not tight for x^S, a contradiction, which completes the proof of the theorem.

A.2. Proof of Theorem 4

1. If $\sum_{i=1}^{p} con(V_i)$ and $|F|$ have the same parity, then inequality (12), and hence inequality (13), would be redundant with respect to the cut and trivial inequalities. Therefore, it cannot be facet defining.

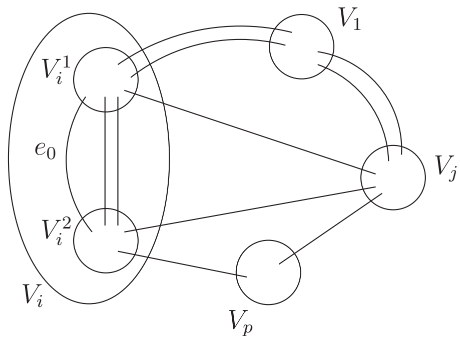

Fig. A1. SP-partition configuration for facet defining.

2. The proof is similar to that of Theorem 2 for Condition (1); hence, it is omitted.
3. If $G(V_i)$ is not 2-edge connected (and connected by 2), let V_i^1, V_i^2 be a partition of V_i such that $[V_i^1, V_i^2] = \{f\}$ for some edge $f \in E$. Let $\pi' = (V_1, \ldots, V_{i-1}, V_i^1, V_i^2, V_{i+1}, \ldots, V_p)$. Suppose that $con(V_i^1) = con(V_i)$ and, by contradiction, $con(V_i^2) \geq 2$. Inequality (12) for π' and F can then be written as

$$x(\delta(\pi') \setminus F) \geq \left\lceil \frac{\sum_{i=1}^{p} con(V_i) + con(V_i^2) - |F|}{2} \right\rceil$$

$$\geq \left\lceil \frac{\sum_{i=1}^{p} con(V_i) - |F|}{2} \right\rceil + 1.$$

As $\delta(\pi') \setminus F = (\delta(\pi) \setminus F) \cup \{f\}$, it follows that inequality (13) (for partition π and F) can be obtained as a combination of the above inequality for π' and the trivial inequality $-x(f) \geq -1$. Hence, (13) cannot define a facet.
4. As $|\delta(V_i)| = 2$, $con(V_i) \leq 2$. Suppose that $con(V_i) = 2$. If $\delta(V_i) = \{e_1, e_2\}$, then it follows that $x(e_1) = x(e_2) = 1$ for every solution x of ESNDP(G). Consider the partition π^1 obtained from π by merging V_i and V_{i+1}. Inequality (12) for π^1 is given by

$$x(\delta(\pi') \setminus F) \geq \left\lceil \frac{\sum_{i=1}^{p} con(V_i) - 2 - |F|}{2} \right\rceil$$

$$= \left\lceil \frac{\sum_{i=1}^{p} con(V_i) - |F|}{2} \right\rceil - 1.$$

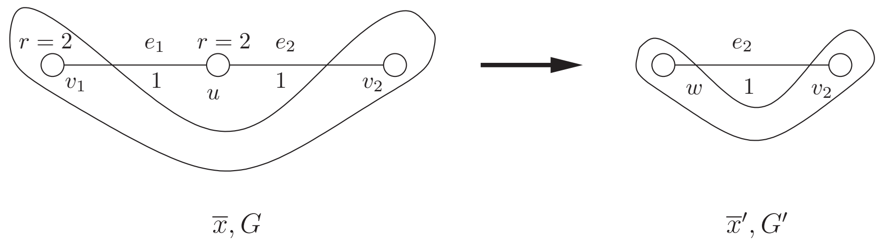

Fig. A2. Operation θ_2.

By adding $x(e_2) = 1$ to the above inequality, we get inequality (12). Hence the latter cannot define a facet.

5. Assume the contrary. Then $p_1(\pi) \geq 2$ and there is an edge between two V_i's of connectivity 1, say V_1 and V_2. As $|F \cap \delta(V_i)| = 1$ for $i = 1, \ldots, p$, and hence $|F| = p$, by Claim 1, inequality (13) can be written as

$$x(\delta(\pi) \setminus F) \geq p_3(\pi) + \left\lceil \frac{p_2(\pi)}{2} \right\rceil. \tag{A4}$$

Suppose that $con(V_i) = 1$ for $i = 1, \ldots, p_1(\pi)$.
Let $\pi' = \left(V'_0, V'_1, \ldots, V'_{p_2(\pi)+p_3(\pi)}\right)$ given by $V'_0 = V_0 \cup V_1 \cup \cdots \cup V_{p_1(\pi)}$ and $V'_i = V_{p_1(\pi)+i}$, $i = 1, \ldots, p_2(\pi) + p_3(\pi)$.
Note that π' has no subsets of connectivity ≤ 1.
Let F' be the subset of edges of F incident to the sets V_i of connectivity ≥ 2. It follows from (A4) that inequality (13), induced by π' and F', is given by

$$\begin{aligned} x(\delta(\pi') \setminus F') &\geq p_3(\pi') + \left\lceil \tfrac{p_2(\pi')}{2} \right\rceil \\ &= p_3(\pi) + \left\lceil \tfrac{p_2(\pi)}{2} \right\rceil. \end{aligned} \tag{A5}$$

Since the V'_i's do not induce a stable set, as $p_2(\pi') = p_2(\pi)$, $p_3(\pi') = p_3(\pi)$, and $p_1(\pi') = 0$, $\delta(\pi') \setminus F'$ is a strict subset of $\delta(\pi) \setminus F$. As $x(e) \geq 0$ for all $e \in E$, we have that inequality (A4) is dominated by inequality (A5), and therefore the former cannot define a facet.

A.3. Proof of Theorem 7

We will show that each of the operations $\theta_1, \theta_2, \theta_3$ preserves the extremality of $\overline{x}'$. This is easily seen in Operation θ_1.

Consider Operation θ_2 (see Fig. A2).

We will show that any cut $\delta(W)$ of $\mathcal{C}(x)$ in G remains tight for $\overline{x}'$ in G'. Clearly, this trivially holds if $\delta(W)$ does not contain e_1. Now suppose that $e_1 \in \delta(W)$.

First, consider the case when both v_1 and v_2 (v_1 and v_2 are the neighbors of u) are in the same shore of $\delta(W)$, say $V \setminus W$. Then $u \in W$ (see Fig. A3). If $con(W) = 2$, then $\overline{x}(\delta(W)) = 2$.

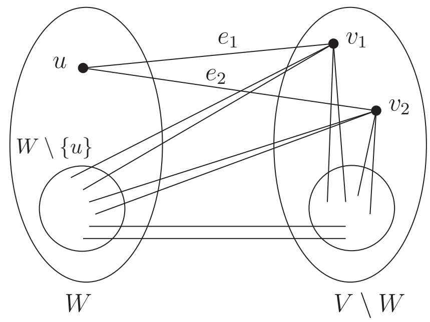

Fig. A3. v_1 and v_2 in the same shore of $\delta(W)$.

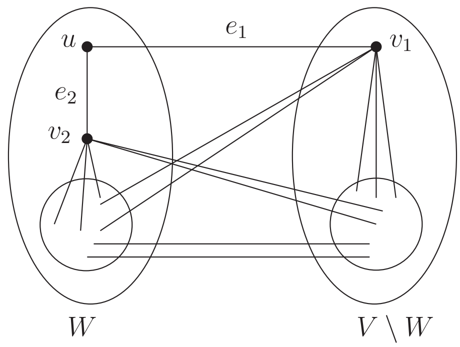

Fig. A4. v_1 and v_2 are not on the same shore of $\delta(W)$.

But here $x(\delta(W)) = 2$ is redundant with respect to $x(e_1) = 1$ and $x(e_2) = 1$, contradicting the fact that $\delta(W) \in \mathcal{C}(x)$.

So suppose that $con(W) = 3$. As $r(u) = 2$, it follows that $W \setminus \{u\} \neq \emptyset$, $r(W \setminus \{u\}) = 3$ and $r(V \setminus W) = 3$.

Let $W' = W \setminus \{u\}$. Clearly $con(W') = 3$. However,

$$\begin{aligned} \overline{x}(\delta(W')) &= \overline{x}(\delta(W)) - \overline{x}(e_1) - \overline{x}(e_2) \\ &= 3 - 1 - 1 \\ &= 1, \end{aligned}$$

which is impossible.

In consequence, v_1 and v_2 should be so that v_1 is in one shore, say $V \setminus W$ and v_2 in the other one (see Fig. A4).

Suppose, on the contrary, that $\overline{x}'$ is not an extreme point of $P(G')$. Then there are two solutions y_1, y_2 of $P(G')$ such that $y_1 \neq y_2$ and $\overline{x}' = \frac{1}{2}(y_1 + y_2)$. Moreover, any inequality of $P(G')$ tight for $\overline{x}'$ is also tight for both y_1 and y_2. Thus $y_1(e) = y_2(e) = 0$ for all $e \in E_0(\overline{x})$, and $y_1(e) = y_2(e) = 1$ for all $e \in E_1(\overline{x}) \setminus \{e_1\}$.

We will discuss the case when $con(W) = 2$. The case when $con(W) = 3$ is similar. We have the following.

Claim A1. *The cut $\delta(W \setminus \{u\})$ is tight for $\overline{x}$.*

Proof. We have that $\delta(W) = \{e_1\} \cup [W \setminus \{u\}, V \setminus W]$. As $\overline{x}(\delta(W)) = 2$ and $\overline{x}(e_1) = 1$, it follows that $\overline{x}[W \setminus \{u\}, V \setminus W] = 1$. On the other hand, $\delta(W \setminus \{u\}) = \{e_2\} \cup [W \setminus \{u\}, V \setminus W]$. Since $\overline{x}(e_2) = \overline{x}(x_1) = 1$, we have that $\overline{x}(\delta(W \setminus \{u\})) = 2$. □

Let $\overline{y}_1 \in \mathbb{R}^E$ be the solution given by

$$\begin{aligned} &\overline{y}_1(e) = y_1(e) && \text{if } e \in E \setminus \{e_1\}, \\ &\overline{y}_1(e) = 1 && \text{if } e = e_1. \end{aligned}$$

We will show that $\overline{y}_1$ is also a solution of $S(\overline{x})$.

It is clear that $\overline{y}_1(e) = 1$ for all $e \in E_1(\overline{x})$, and $\overline{y}_1(e) = 0$ for all $e \in E_0(\overline{x})$. In particular, we have that $\overline{y}_1(e_2) = 1$.

Now consider a cut $\delta(W) \in \mathcal{C}(\overline{x})$. If e_1 does not belong to $\delta(W)$, then clearly $\delta(W)$ is also tight for $\overline{x}$ and therefore for y_1 and $\overline{y}_1$. If $e_1 \in \delta(W)$, then by Claim A1, $\delta(W \setminus \{u\})$ is also tight for $\overline{x}$, and hence for y_1. Thus $\delta(W \setminus \{u\})$ is tight for $\overline{y}_1$. Since $\overline{y}_1(e_1) = 1$, we obtain that $\overline{y}_1(\delta(w)) = \overline{y}_1(\delta(W \setminus \{u\})) - \overline{y}_1(e_2) + \overline{y}_1(e_1) = 2 - 1 + 1 = 2$. Consequently, $\overline{y}_1$ is a solution of $S(\overline{x})$.

However, since $\overline{x}' = \frac{1}{2}(y_1 + y_2)$ and $y_1 \neq y_2$, it follows that $y_1 \neq \overline{x}$. But this is impossible since $\overline{x}$ is the unique solution of $S(\overline{x})$.

Now, suppose that $G' = (V', E')$ is obtained by one application of operation θ_3.

Note that any cut $\delta(U) \in \mathcal{C}(\overline{x})$ does not intersect $E(W)$. For otherwise, as $G(W)$ is k-edge connected and $\overline{x}(e) = 1$ for all $e \in E(W)$, $x(\delta(U)) = con(U)$ would be redundant with respect to the equations $x(e) = 1$, $e \in E(W)$, which is a contradiction.

Let S' be the system obtained from $S(\overline{x})$ by removing all the equations $x(e) = 1$ for all $e \in E(W)$. Obviously, S' is nonsingular. As $\overline{x}'$ is a solution of S', it is then a unique solution and therefore an extreme point of $\mathrm{P}(G')$, which completes the proof of the theorem.