Ann Oper Res (2006) 146:41–73
DOI 10.1007/s10479-006-0047-2

Design of survivable IP-over-optical networks

Sylvie Borne · Eric Gourdin · Bernard Liau · A. Ridha Mahjoub

Published online: 27 June 2006

Abstract In the past years, telecommunications networks have seen an important evolution with the advances in optical technologies and the explosive growth of the Internet. Several optical systems allow a very large transport capacity, and data traffic has dramatically increased. Telecommunications networks are now moving towards a model of high-speed routers interconnected by intelligent optical core networks. Moreover, there is a general consensus that the control plan of the optical networks should utilize IP-based protocols for dynamic provisioning and restoration of lightpaths. The interaction of the IP routers with the optical core networks permits to achieve end-to-end connections, and the lightpaths of the optical networks define the topology of the IP network. This new infrastructure has to be sufficiently survivable, so that network services can be restored in the event of a catastrophic failure. In this paper we consider a multilayer survivable network design problem that may be of practical interest for IP-over-optical neworks. We give an integer programming formulation for this problem and discuss the associated polytope. We describe some valid inequalities and study when these are facet defining. We discuss separation algorithms for these inequalities

S. Borne (✉)
Laboratoire LIMOS, CNRS, Université Blaise Pascal - Clermont Ferrand II,
Complexe scientifique des Cézeaux, 63177 Aubière Cedex, France
e-mail: sylvie.borne@isima.fr

E. Gourdin
Laboratoire CORE/CPN, France Télécom div. R&D,
38-40 rue du Général-Leclerc, 92794 Issy-les-Moulineaux Cedex 9, France
e-mail: eric.gourdin@rd.francetelecom.com

B. Liau
Laboratoire ROSI/DCAS, France Télécom Immeuble Bertrand, 6 place d'Alleray,
75505 Paris cedex 15, France
e-mail: bernard.liau@francetelecom.com

A. R. Mahjoub
Laboratoire LIMOS, CNRS, Université Blaise Pascal-Clermont Ferrand II,
Complexe scientifique des Cézeaux, 63177 Aubière Cedex, France
e-mail: Ridha.Mahjoub@math.univ-bpclermont.fr

and introduce some reduction operations. We develop a Branch-and-Cut algorithm based on these results and present extensive computational results.

Keywords IP-over-optical network · Survivability · Integer programming · Branch-and-Cut algorithm

1. Introduction

In the past years, telecommunications networks have seen a big development with the advances in optical technologies and the explosive growth of the Internet. Also the data traffic has increased dramatically and has now surpassed voice traffic in volume. Using the new optical technologies, different systems allow a very large increase of transport capacity and the transfert of almost illimitated quantities of informations. Hence, in the event of a catastrophic failure, a big amount of traffic may be lost. Now telecommunications networks must have a survivable topology, that is to say a topology that permits to the service to be restored and the network to remain functionnal in the event of a failure. For this network survivability has become a major objective in the design of telecommunications networks.

Data networks have always been analysed, described and managed in a multilayer structure. Indeed, it is quite natural to assume that the more elaborate functionalities of a network rely on a set of simple ones provided by some lower layer. This is in particular the case of modern telecommunications networks where different technologies (SDH/SONET, WDM, Gigabit Ethernet, ATM, IP, . . .) are combined in various ways on successive layers. From a practical point of view, this means that, in order to carry its traffic on some layer, the network may need to use a lower-level technology. Then several layers can be piled up in order to have an operational network offering a variety of services. The advantage of this is that each technology can be used for its most favorable features. Moreover, each technology is characterized by a certain range of traffic rates. The drawback, however, is that each technology, and hence each layer, manages its own routing control scheme independantly from the others, and addresses its own survivability issues.

The capacities of a given layer correspond to the (worst-case) traffic demands that must be routed on the layer just below. The process of determining the capacities (usually called *dimensioning*) to install on the different layers of a network often reduces to a succession of multicommodity flow problems. Usually there is an empirical relation between these problems, and the whole dimensioning problem is never treated in an optimal way. As a consequence, in a network design problem, reliability is considered layer by layer without tackling the redondancy and the non-optimality yielded by the multilayer structure. Moreover, a failure in the network can be handled by several successive layers. This results in a potential huge global over-provisioning of ressources, each layer protecting in turn the ones above. However the relation between technologies used in the different layers is usually complex, and does not permit to efficiently correlate the control of the successive layers. In consequence, the solution provided for this multilayer survivability problem usually consists of an over protection of the whole network. But this may be very costly and sometimes not efficient.

The introduction of new protocols in telecommunications (like GMPLS (Zouganeli, 2001)) gives a new trend for multilayer data networks. This new system provides a common signaling and routing framework between the different layers, and it does not restrict the way these layers work together. This evolution is yielding new survivability issues in multilayer networks. In this paper we introduce a multilayer survivable network design problem that may be of practical interest for the design of survivable IP-over-optical networks, the networks that consist of two layers, the IP (service, client) layer and the optical (transport) layer. We give an integer programming formulation for this problem and devise a Branch-and-Cut algorithm.

Survivability and dimensioning have been already studied in the literature for multilayer networks. In particular, heuristic approaches have been proposed. In Gouveia et al. (2003), Gouveia and Patrício study the design of MPLS-over-WDM networks. They address the dimensioning subject to some path constraints in the WDM layer and hop constraints in the MPLS layer. They give an integer programming formulation and devise a heuristic technique based on that formulation. In Ricciato et al. (2002), Ricciato et al. consider the problem of off-line configuration of MPLS-over-WDM networks under time-varying offered traffic. They present a mixed integer programming formulation for the problem and discuss heuristic approaches.

The paper is organised as follows. In the following section we discuss the IP-over-optical networks and examine the interconnection models proposed for these networks as well as system GMPLS and its interaction with these models. In Section 3, we present a multilayer survivable network design problem, called the multilayer survivable IP network design problem, and give a 0–1 integer programming formulation for this problem. In Section 4, we study the associated polytope. We give the dimension of the polytope, identify three classes of valid inequalities and describe conditions for theses inequalities to be facet defining. In Section 5, we introduce some reduction operations. In Section 6, we discuss separation techniques and describe a Branch-and-Cut algorithm for the problem. Our computational results are presented and discussed in Section 7. In Section 8, we give some concluding remarks.

2. A new infrastructure of telecommunications networks

Telecommunications networks are now moving toward a model of high-speed routers interconnected by intelligent optical core networks. Moreover, there is a general consensus that the control plan of the optical networks should utilize IP-based protocols for dynamic provisioning and restoration of lightpaths (Bradner, 1995; Jensen, 2001; Postel, 1981a, 1981b, 1981c).

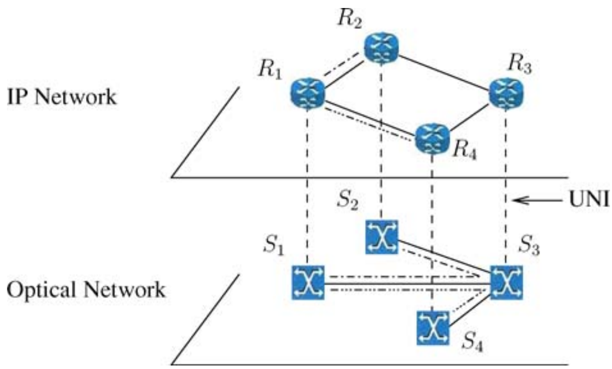
The optical network consists of multiple switches (also called Optical Cross-Connects (OXC)) interconnected by optical links. The IP and optical networks communicate through logical control interfaces called User-Network-Interfaces (UNI). The optical network essentially provides point-to-point connectivity between routers in the form of fixed bandwidth lightpaths. These lightpaths define the topology of the IP network.

Each router in the IP network is connected to at least one of the optical switches. Moreover to each link in the IP network between two routers corresponds a routing path in the optical one between two switches corresponding to these routers. Figure 1 shows an IP-over-optical network. The IP network has four routers $R_1, \ldots, R_4$ and the optical network has four switches $S_1, \ldots, S_4$. Each optical switch communicates with one router throught the UNI.

In order to formalize the IP-over-optical network, the Internet Engineering Task Force (IETF) proposed three interconnection models (Rajagopalan et al., 2000) : overlay, peer and augmented models.

The overlay model is the model currently used by the operators. Under this model, the IP network routing, topology distribution, and signaling protocols are independent of the corresponding protocols in the optical network. The client network requests a connection between two routers. The optical network offers end-to-end wavelength services to client network via the UNI to respond to this request. In this model, the client network has otherwise no control over the exact routing and priority received within the intelligent optical network

Fig. 1 An IP-over-optical network



which has full control over its network ressources. The advantage of the overlay model is that it is the most practical for near-term deployment. Its drawback is that it requires the creation and management of IP routing adjacencies over the optical network.

Under the peer model, the IP and optical networks are treated together as a single integrated network managed and traffic engineered in a unified manner. In this regard, the optical switches are treated just like any other IP router as far as the control plane is concerned. IP routers and optical switches use the same addressing scheme. The optical network elements become IP addressable entities. The optical network topology is fully visible to routers. A single routing protocol instance runs over both the IP and optical domains. The advantage of the peer model is that it allows seamless interconnection of IP and optical networks. The architecture is scalable, functionality is not duplicated and conflicts between several control planes do not arise. Its drawback is that it requires routing information specific to optical networks to be known to IP routers. There are excessive information flows between the two networks. This type of tight integration may not be pratical in the near term. Despite its drawbacks, the peer model can be expected to be the architecture to be adopted in the long term if IP indeed dominates the scene.

The augmented model is between the overlay and the peer models. There are separate routing instances of the same routing protocol in the IP and optical domains, but with limited routing exchange between the two domains. Some reachability information is exchanged between the two networks but the topology of the optical network is opaque to the client network. This option may be a good compromise in that it is relatively easy to deploy in the near term compared to the peer model, and at the same time it is less rigid and more efficient than the overlay model.

The introduction of the peer model (and the GMPLS control plane) gives rise to new survivability issues for the IP-over-optical networks. For example consider the IP-over-optical network given in Figure 1. Suppose that the link $R_1 - R_2$ of the IP network, corresponds to the optical path $S_1 - S_3 - S_2$, and the link $R_1 - R_4$ corresponds to the path $S_1 - S_3 - S_4$. Here, the network is not survivable. For instance, if the optical link $S_1 - S_3$ fails, then the links in the IP network $R_1 - R_2$ and $R_1 - R_4$ are cut, and therefore the router R_1 is no more connected to the rest of the routers. In consequence, survivability strategies have to be considered. If the transport network is fixed, one has to determine the suitable client network topology for the network to be survivable. If however, the client network is fixed as well as its routing, then we have to determine an optimal survivable routing in the transport network. And finally if none of both layers is fixed, then a routing has to be determined for each of the layers in order to get a whole survivable network. In this paper we shall be interested in the first problem.

3. The multilayer survivable IP network design problem

3.1. The problem

The first major survivability requirement used in telecommunications networks is the so-called *2-connectivity*. That is there must exist at least two edge-disjoint paths between every pair of nodes in the network. This implies that the network remains connected in the event of any single edge failure. The problem of finding a minimum cost 2-edge connected subgraph has been extensively investigated in the past decade (Barahona and Mahjoub, 1995; Grötschel, Monma, and Stoer, 1995; Kerivin, 2000; Kerivin, Mahjoub, and Nocq, 2004; Mahjoub, 1994; Stoer, 1992).

The assumption that only one edge may fail at a time is based on the naive idea that the links in the network are independant and no equipment can be commonly used by two distinct links. However, this is not the case, for instance, for the IP-over-optical networks, when the optical layer is taken into account in the management of the IP network.

In fact, any edge of the client network is supported by a path in the optical network (lightpaths). That is the traffic of an edge in the client network is routed in the optical network along the path corresponding to that edge. Therefore an edge of the optical network may appear in several paths supporting distinct edges. In consequence, the failure of an edge in the optical network may affect several optical paths, and hence the edges of the client network corresponding to these paths. Moreover, these edges may all fail at the same time.

Consequently, a more realistic model which has to be investigated, would consist in designing a minimum cost client network that remains connected for any simple edge failure in the transport network.

In this section we consider this problem. More precisely we consider the overlay model where the IP and optical networks are separated. We suppose that the topology and the routing of the optical network are fixed and satisfy some survivability requirements. We suppose that a set of IP routers (resp. optical switches) is given as well as the possible links between the routers (resp. switches). As the routing of the optical network is known, one can determine for each optical link e, the set of edges of the IP network that may be affected if e is cut. If a certain cost is associated with each edge of the IP network, the *multilayer survivable IP network design problem* (MSIPND problem) is to find the set of links to be installed in the IP network so that if a failure occurs on an optical link, the IP subnetwork obtained by removing the corresponding edges is connected.

In what follows we give an integer programming formulation for the MSIPND problem. To this end, we first introduce some definitions and notations.

3.2. Definitions and notations

We consider undirected and finite graphs. We denote a graph by $G = (V, E)$ where V is the *node set* and E the *edge set* of G. If $e \in E$ is an edge between two nodes u and v, then we also write $e = uv$ to denote e. Given $W \subseteq V$, we denote by $\delta_G(W)$ the set of edges of G having exactly one node in W. The edge set $\delta_G(W)$ is called a *cut*. A subset $F \subseteq E$ of G is called an *edge cutset* if F is a cut. If $W \subset V$, $\overline{W}$ denotes $V \backslash W$. For $F \subseteq E$ we let $G \backslash F$ denote the subgraph of G obtained by removing the edges of F and by $G[F]$ the subgraph of G induced by F. If $F = \{e\}$, we write $G \backslash e$ for $G \backslash \{e\}$. For $W \subseteq V$, we denote by $G(W)$ the

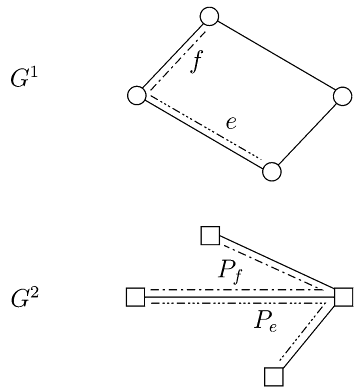


Fig. 2 Graphs of an IP-over-optical network

subgraph of G induced by W. If U and W are two node subsets such that $U \cap W = \emptyset$, then we denote by $[U, W]$ the set of edges having one node in U and the other in W. If $V_1, \ldots, V_p$ is a partition of V, we let $\delta_G(V_1, \ldots, V_p)$ denote the set of edges of G between the elements of the partition. For $F \subset E$, $V_G(F)$ denotes the set of nodes of the edges of F. For $W \subset V$, $E_G(W)$ denotes the set of edges of E having both endnodes in W.

Given a graph $G = (V, E)$, *contracting* an edge set F of G consists in contracting the nodes of $V_G(F)$ to a new node w (retaining parallel edges). A *path* P in $G = (V, E)$ is an alternate sequence of nodes and edges $(v_1, e_1, v_2, e_2, \ldots, v_p, e_p, v_{p+1})$ such that $e_i = v_i v_{i+1}$ for $i = 1, \ldots, p$ and $v_i \neq v_j$ for $i = 1, \ldots, p+1$, $j = 1, \ldots, p+1$. Nodes v_1, v_{p+1} are the extremities of P and we will say that P goes from v_1 to v_{p+1} or P is between v_1 and v_{p+1}. If no confusion may arise, we will sometimes denote P by either its sequence of edges $(e_1, \ldots, e_p)$ or its sequence of nodes $(v_1, \ldots, v_{p+1})$. A *chord* of P is an edge of G that is not in P and that connects two nodes of P. A path P is called a *cycle* if $v_{p+1} = v_1$, $p \geq 1$, $v_1, \ldots, v_{p+1}$ are all distinct, and $e_1, \ldots, e_p$ are all distincts.

Given a vector $x \in I\!R^E$ and $F \subseteq E$, we let $x(F) = \sum_{e \in F} x(e)$.

Throughout the paper, given an IP-over-optical network, we suppose that to each router of the IP layer corresponds exactly one optical switch. We will represent an IP-over-optical network by two graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$, that represent the IP and optical networks, respectively. The nodes of G^1 (resp. G^2) correspond to the routers of the IP layer (resp. the optical switches), and the edges represent the possible links between the routers (resp. switches). For an edge $f \in E^1$, we denote by P_f the path in G^2 corresponding to f. Figure 2 shows graphs G^1 and G^2 corresponding to the IP-over-optical network of Fig. 1. In G^2, are indicated two paths P_e and P_f which correspond to the edges e and f of G^1.

3.3. Formulation

In terms of graphs, the MSIPND problem can be presented as follows. Let $c : E^1 \longrightarrow I\!R_+$ be a function that associates with each edge f of graph $G^1 = (V^1, E^1)$, a cost $c(f) > 0$. For an edge e of graph $G^2 = (V^2, E^2)$ corresponding to the optical network, let F_e be the set of edges of the IP network that may be affected by a failure of e, that is $F_e = \{f \in E^1 \mid e \in P_f\}$. Then, the MSIPND problem consists in finding a minimum weight subgraph H of G^1 such that for every edge $e \in E^2$, the graph obtained from H by removing the edges of F_e is connected.

Note that if $|F_e| = 1$ for all $e \in E^2$ and $\bigcup_{e \in E^2} F_e = E^1$, the MSIPND problem is nothing but the 2-edge connected subgraph problem. As this latter problem is NP-hard, the MSIPND problem so is.

In order to formulate this problem, let us associate with each $F \subseteq E^1$, the incidence vector x^F given by $x^F(f) = 1$ if $f \in F$ and $x^F(f) = 0$ otherwise. It is not hard to see that the MSIPND problem is equivalent to the following integer programming problem:

$$\text{Minimize} \sum_{f \in E^1} c(f)x(f)$$

$$x(\delta_{G^1 \setminus F_e}(W)) \geq 1 \qquad \text{for all } W \subseteq V^1, \quad \emptyset \neq W \neq V^1, \quad \text{for all } e \in E^2, \tag{1}$$

$$0 \leq x(f) \leq 1 \qquad \text{for all } f \in E^1, \tag{2}$$

$$x(f) \in \{0, 1\} \qquad \text{for all } f \in E^1. \tag{3}$$

Inequalities (1) express the fact that G^1 remains connected after removing the edges of F_e, for all $e \in E^2$. They will be called *cut inequalities*. Inequalities (2) are called *trivial inequalities*.

In the MSIPND problem we suppose that the failures may happen only on the links of the optical network. We may also consider the case when the failures may affect the optical switches as well. For this, one can associate with each node $v \in V^2$ the edge set $F_v = \bigcup_{e \in \delta(v)} F_e$. The problem here would be to find a subgraph H such that for each $e \in E^2$ ($v \in V^2$), the subgraph of H obtained by removing F_e (F_v) is connected. The problem in this case can be formulated in a similar way.

4. Associated polyhedron and valid inequalities

In this section we will discuss the polytope associated with the MSIPND problem. We will describe its dimension and identify three classes of valid inequalities.

Throughout the two following sections we consider a graph $G = (V, E)$ and a family $\mathcal{F} = \{F_1, \ldots, F_t\} \subseteq 2^E$ with $t \geq 2$ of edge subsets of E. For $i \in \{1, \ldots, t\}$, we will denote by $G_i = (V, E_i)$ the subgraph of G obtained by removing the edges of F_i. Hence $E_i = E \setminus F_i$. Let MSIPND($G, \mathcal{F}$) denote the convex hull of the integer solutions of the system

$$x(\delta_{G_i}(W)) \geq 1 \qquad \text{for all } W \subseteq V, \quad \emptyset \neq W \neq V, \quad i = 1, \ldots, t, \tag{4}$$

$$0 \leq x(f) \leq 1 \qquad \text{for all } f \in E. \tag{5}$$

Note that if $G = G^1$ and $\mathcal{F} = \{F_e,\ e \in E^2\}$, MSIPND($G, \mathcal{F}$) is nothing but the polytope associated with the MSIPND problem. MSIPND($G, \mathcal{F}$) will then be called the *Multilayer Survivable IP Network Design Polytope*. We will assume that a solution of the MSIPND problem is a set of edges $T \subseteq E$ such that $G[T \setminus F_i]$ is connected for $i = 1, \ldots, t$. We will denote by $\mathcal{S}(G, \mathcal{F})$ the set of solutions of the MSIPND problem with respect to G and $\mathcal{F}$. We will also assume that $E \in \mathcal{S}(G, \mathcal{F})$.

An edge e of G is said to be *essential* if there is $i \in \{1, \ldots, t\}$ such that e is an edge cutset of G_i. Note that e is essential if and only if e belongs to every solution $T \in \mathcal{S}(G, \mathcal{F})$. Let E^* be the set of essential edges of G. The following theorem gives the dimension of the polytope.

Theorem 4.1. *dim(MSIPND($G, \mathcal{F}$))* $= |E| - |E^*|$.

Proof: If $e \in E^*$, as $e \in T$ for all $T \in \mathcal{S}(G, \mathcal{F})$, the constraint $x(e) = 1$ is an equation of the linear system describing MSIPND$(G, \mathcal{F})$. Hence dim(MSIPND$(G, \mathcal{F})) \leq |E| - |E^*|$. On the other hand, if $e \in E \backslash E^*$, $T_e = E \backslash \{e\} \in \mathcal{S}(G, \mathcal{F})$. By considering the sets T_e, $e \in E \backslash E^*$ and E, we have that the incidence vectors x^{T_e}, $e \in E \backslash E^*$, x^E are affinelly independent. Thus dim(MSIPND$(G, \mathcal{F})) \geq |E| - |E^*|$, and therefore the theorem follows. □

As a consequence of Theorem 4.1, we have the following corollary.

Corollary 4.2. *MSIPND$(G, \mathcal{F})$ is full dimensional if and only if G_i is 2-edge connected for $i = 1, \ldots, t$.*

In the remainder of this section we assume that MSIPND$(G, \mathcal{F})$ is full dimensional. We also assume that $G = (V, E)$ is a complete graph. This latter assumption is not restrictive since the problem in an incomplete graph can be reduced to the problem in a complete graph by giving sufficiently high cost to non-existing edges. We also suppose that every edge of G belongs to some $F_i \in \mathcal{F}$. In the following, we introduce three classes of valid inequalities. We also give necessary conditions and sufficient conditions for these inequalities to be facet defining. We assume that the reader is familiar with polyhedral combinatorics, for more details see Schrijver (2003).

A subgraph $H = (W, F)$ of $G = (V, E)$ is said to be *$\mathcal{F}$-connected* with respect to $\mathcal{F} = \{F_1, \ldots, F_t\}$ if for all $i \in \{1, \ldots, t\}$, the graph $H \backslash F_i$ is connected. H is said to be *2-$\mathcal{F}$-connected* if for all $e \in F$, the graph $H \backslash e$ is $\mathcal{F}$-connected.

4.1. Partition inequalities

Proposition 4.3. *Let $V_1, \ldots, V_p$, $p \geq 2$, be a partition of V and $F_i \in \mathcal{F}$. Then*

$$x(\delta_{G_i}(V_1, \ldots, V_p)) \geq p - 1 \tag{6}$$

is valid for MSIPND$(G, \mathcal{F})$.

Proof: Since G_i must be connected, every graph obtained from G_i by contraction of edges must also be connected, and hence contains a spanning tree. If the number of nodes of the contracted graph is p, then this graph contains at least $p - 1$ edges. □

Inequalities (6) are called *partition inequalities*.

If $G = (V, E)$ is a graph and $V_1, \ldots, V_p$, $p \geq 2$, is a partition of V, we denote by $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ the subgraph of G obtained by contracting the nodes of V_j, $j = 1, \ldots, p$. Hence $\widetilde{G}_i = (\widetilde{V}, \widetilde{E}_i)$ will denote the graph obtained from G_i, $i = 1, \ldots, t$, by contracting the sets $V_1, \ldots, V_p$. We also denote by $\widetilde{F}_i$ the restriction of F_i in $\widetilde{E}$.

Theorem 4.4. *Inequality (6) defines a facet of MSIPND$(G, \mathcal{F})$ only if*

(a) *$G_i(V_j)$ is 2-edge connected, for $j = 1, \ldots, p$,*

(b) *$G(V_j)$ is $\mathcal{F}$-connected for $j = 1, \ldots, p$, if $\widetilde{E} \cap F_i = \emptyset$,*

(c) *for every $j \in \{1, \ldots, t\} \backslash \{i\}$ such that $F_j \cap \widetilde{E}_i \neq \emptyset$, there is an edge set $\widetilde{T} \subseteq \widetilde{E}_i$ such that*

 (c.1) *$F_j \cap \widetilde{T} \neq \emptyset$,*

 (c.2) *$\widetilde{G}_i[\widetilde{T}]$ is a tree and,*

 (c.3) *any cut $\delta_{\widetilde{G}}(W)$ intersects F_i, if $W \subset \widetilde{V}$ and $\delta_{\widetilde{G}_i}(W) = F_j \cap \widetilde{T}$,*

(d) *there exists an edge set* $\widetilde{T} \subset \widetilde{E}_i$ *that induces a tree in* $\widetilde{G}_i$*, and such that*

$$T_0 = \widetilde{T} \cup F_i \cup \left(\cup_{j=1}^{p} E_G(V_j) \right)$$

is a solution of $\mathcal{S}(G, \mathcal{F})$*,*

Proof:

(a) Suppose for instance that $G_i(V_1)$ is not 2-edge connected. Hence there is a partition V_1^1, V_1^2 of V_1 such that $|[V_1^1, V_1^2] \backslash F_i| \leq 1$. Consider the partition $V_1', \ldots, V_{p+1}'$ such that

$$\begin{aligned} V_1' &= V_1^1, \\ V_2' &= V_1^2, \\ V_j' &= V_{j-1} \quad \text{for } j = 3, \ldots, p+1. \end{aligned}$$

If $[V_1^1, V_1^2] \backslash F_i = \emptyset$, then inequality (6) is dominated by the partition inequality corresponding to partition $V_1', \ldots, V_{p+1}'$. If this is not the case and, hence $[V_1^1, V_1^2] \backslash F_i = \{f\}$ for some edge f, then inequality (6) can be written as the sum of $-x(f) \geq -1$ and the partition inequality associated with $V_1', \ldots, V_{p+1}'$.

(b) Suppose that $G(V_1)$ is not $\mathcal{F}$-connected. Then there is $k \in \{1, \ldots, t\}$ such that $G_k(V_1)$ is not connected. In consequence, there is a partition V_1^1, V_1^2 of V_1 with $[V_1^1, V_1^2] \backslash F_k = \emptyset$. If $\widetilde{E} \cap F_i = \emptyset$, then any solution of $\mathcal{S}(G, \mathcal{F})$ must contain at least p edges from $\widetilde{E}_i$. This implies that inequality (6) is satisfied with strict inequality by the incidence vector of any solution of $\mathcal{S}(G, \mathcal{F})$. Therefore (6) can not define a facet.

(c) Assume the contrary. Then let $j \in \{1, \ldots, p\} \backslash \{i\}$ with $F_j \cap \widetilde{E}_i \neq \emptyset$ such that for every $\widetilde{T} \subseteq \widetilde{E}_i$, at least one of the statements (c.1) (c.2) and (c.3) is not satisfied. We will show that (6) can not define a facet. As $F_j \cap \widetilde{E}_i \neq \emptyset$, let $f \in F_j \cap \widetilde{E}_i$. If inequality (6) defines a facet, as (6) is different from the inequalities $x(f) \geq 0$, there must exist a solution, say $T \in \mathcal{S}(G, \mathcal{F})$ containing f whose incidence vector satisfies inequality (6) as equation. As $G[T \backslash F_i]$ is connected, $\widetilde{T} = T \cap \widetilde{E}_i$ induces a connected subgraph of $\widetilde{G}_i$. Moreover as $|\widetilde{T}| = |T \cap \widetilde{E}_i| = p - 1$, $\widetilde{T}$ must be a tree in $\widetilde{G}_i$. Note that $f \in \widetilde{T}$ and hence $\widetilde{T} \cap F_j \neq \emptyset$. Therefore $\widetilde{T}$ satisfies (c.1) and (c.2). By our hypothesis, $\widetilde{T}$ does not thus satisfy (c.3). Let $W \subseteq \widetilde{V}$ be a node set that induces the unique cut $\delta_{\widetilde{G}_i}(W)$ in $\widetilde{G}_i$ such that $\delta_{\widetilde{G}_i}(W) = \widetilde{T} \cap F_j$. (This cut can be detected by a simple labeling of the nodes.) As $\widetilde{T}$ does not satisfy (c.3), it follows that $\delta_{\widetilde{G}}(W) \cap F_i = \emptyset$. Since $\delta_{\widetilde{G}_i}(W) = \widetilde{T} \cap F_j$, we have that $\delta_{\widetilde{G}}(W) \cap T \subseteq F_j$, and therefore the subgraph induced by $T \backslash F_j$ is not connected. But this contradicts the fact that T is a solution of $\mathcal{S}(G, \mathcal{F})$.

(d) If the statement does not hold, then every solution of $\mathcal{S}(G, \mathcal{F})$ contains at least p edges from $\widetilde{E}_i$. Hence the face defined by inequality (6) is empty, and therefore this inequality can not define a facet.

□

Theorem 4.5. *Inequality* (6) *defines a facet of MSIPND*$(G, \mathcal{F})$ *if*

(a) *condition* (a), (b), (c) *of Theorem* (4.4) *are satisfied,*

(b) $G(V_i)$ *is 2-*$\mathcal{F}$*-connected for* $i = 1, \ldots, p$*,*

(c) *there is* $\widetilde{T} \subseteq \widetilde{E}_i$ *satisfying condition* (d) *of Theorem* 4.4*, such that for all* $W \subset \widetilde{V}$ *and* F_j*,* $j \neq i$*, if* $\delta_{\widetilde{G}}(W) \cap (\widetilde{F}_i \backslash \widetilde{F}_j) = \emptyset$ *(resp.* $|\delta_{\widetilde{G}}(W) \cap (\widetilde{F}_i \backslash \widetilde{F}_j)| = 1$*), then* $|\delta_{\widetilde{G}}(W) \cap (\widetilde{T} \backslash \widetilde{F}_j)| \geq 2$ *(resp.* $|\delta_{\widetilde{G}}(W) \cap (\widetilde{T} \backslash \widetilde{F}_j)| \geq 1$*),*

(d) $\widetilde{G}_i$ *is* 2*-node connected.*

Proof: Let us denote inequality (6) by $ax \geq \alpha$, and let $bx \geq \beta$ be a facet defining inequality of MSIPND$(G, \mathcal{F})$ such that $\{x \in$ MSIPND$(G, \mathcal{F}) | ax = \alpha\} \subseteq \{x \in$ MSIPND$(G, \mathcal{F}) | bx = \beta\}$. To show that $ax \geq \alpha$ define a facet, it suffices to show that there is $\rho > 0$ such that $b = \rho a$. For this, we first prove that there exists $\rho \in I\!R$ such that

$$b(e) = \rho \quad \text{for all } e \in \widetilde{E}_i. \tag{7}$$

By condition c) there exists an edge set $\widetilde{T} \subseteq \widetilde{E}_i$ inducing a tree in $\widetilde{G}_i$ such that $T_0 = \widetilde{T} \cup F_i \cup (\cup_{j=1,\ldots,p} E_G(V_j))$ belongs to $\mathcal{S}(G, \mathcal{F})$. Let $\widetilde{T}_0 = \widetilde{T} \cup \widetilde{F}_i$. We have that $\widetilde{T}_0$ is a solution of $\mathcal{S}(\widetilde{G}, \widetilde{\mathcal{F}})$. As $\widetilde{T}$ is a spanning tree in $\widetilde{G}_i$, for any edge $e \in \widetilde{E}_i \backslash \widetilde{T}$, there is a unique cycle in $\widetilde{G}_i$ formed by e and a path of $\widetilde{T}$. We will denote by $\mathcal{C}_e$, V_e and P_e, this cycle, its node set and the path of $\widetilde{T}$ in $\mathcal{C}_e$, respectively.

Let $e \in \widetilde{E}_i \backslash \widetilde{T}$. Let $f \in \mathcal{C}_e \backslash \{e\}$. Let $T_e = (T_0 \backslash \{f\}) \cup \{e\}$. We claim that $T_e \in \mathcal{S}(G, \mathcal{F})$. As $T_0 \in \mathcal{S}(G, \mathcal{F})$, it suffices to show that $\widetilde{T}_e = (\widetilde{T}_0 \backslash \{f\}) \cup \{e\}$ is a solution of $\mathcal{S}(\widetilde{G}, \widetilde{\mathcal{F}})$. First of all note that, as $(\widetilde{T} \backslash \{f\}) \cup \{e\}$ is a tree in $\widetilde{G}_i$, $\widetilde{T}_e \backslash \widetilde{F}_i$ induces a connected subgraph in $\widetilde{G}_i$. Now consider a set $\widetilde{F}_j$, $j \neq i$. If $f \in \widetilde{F}_j$, as $\widetilde{T}_0 \in \mathcal{S}(\widetilde{G}, \widetilde{\mathcal{F}})$ and hence $\widetilde{G}[\widetilde{T}_0 \backslash \widetilde{F}_j]$ is connected, it follows that $\widetilde{G}[\widetilde{T}_e \backslash \widetilde{F}_j]$ is connected. So suppose $f \notin \widetilde{F}_j$. If $\widetilde{G}[\widetilde{T}_e \backslash \widetilde{F}_j]$ is not connected, then there is $W \subseteq \widetilde{V}$ such that $\delta_{\widetilde{G}}(W) \cap (\widetilde{T}_e \backslash \widetilde{F}_j) = \emptyset$. As $\widetilde{G}[\widetilde{T}_0 \backslash \widetilde{F}_j]$ is connected, one should have $\delta_{\widetilde{G}}(W) \cap (\widetilde{T}_0 \backslash \widetilde{F}_j) = \{f\}$. Hence $\delta_{\widetilde{G}}(W) \cap (\widetilde{F}_i \backslash \widetilde{F}_j) = \emptyset$ and $|\delta_{\widetilde{G}}(W) \cap (\widetilde{T} \backslash \widetilde{F}_j)| < 2$. But this is a contradiction. Thus $\widetilde{G}[\widetilde{T}_e \backslash \widetilde{F}_j]$ is connected. In consequence, $\widetilde{T}_e \in \mathcal{S}(\widetilde{G}, \widetilde{\mathcal{F}})$, and therefore $T_e \in \mathcal{S}(G, \mathcal{F})$.

As $ax^{T_0} = ax^{T_e} = \alpha$, we have $bx^{T_0} = bx^{T_e} = \beta$ and in consequence $b(e) = b(f)$. As f is an arbitrary edge of $\mathcal{C}_e$ and e is arbitrary in $\widetilde{E}_i \backslash \widetilde{T}$, we have

$$b(e) = b(f) \quad \text{for all } f \in \mathcal{C}_e, \quad \text{for all } e \in \widetilde{E}_i \backslash \widetilde{T}. \tag{8}$$

If $\mathcal{C}_e$ contains a chord, say $h = uv$, not in F_i, then $h \in \widetilde{E}_i \backslash \widetilde{T}$. Note that $\mathcal{C}_h \backslash \{h\} \subset \mathcal{C}_e$. By (8) with respect to e and $\mathcal{C}_e$ and h and $\mathcal{C}_h$ we obtain that $b(e') = \rho$ for some ρ for all edge $e' \in \mathcal{C}_e \cup \mathcal{C}_h$. As h is an arbitrary chord of $\mathcal{C}_e$, we get

$$b(e') = \rho \quad \text{for all } e' \in E_{\widetilde{G}_i}(V_e). \tag{9}$$

Since by d) $\widetilde{G}_i$ is 2-node connected, $\widetilde{E}_i \backslash \widetilde{T} \neq \emptyset$. Hence, by the above development together with (9), we may suppose that there is a node subset $U \subset \widetilde{V}$ such that the restriction of $\widetilde{T}$ on U, say $\widetilde{T}_U$ is a spanning tree in $\widetilde{G}_i(U)$, and

$$b(e) = \rho \quad \text{for all } e \in E_{\widetilde{G}_i}(U). \tag{10}$$

If $U = \widetilde{V}$, then (7) is satisfied and we are done. Suppose this is not the case. So to prove (7), it suffices to show that there is a subset U' that strictly contains U and for which (10) holds. As $\widetilde{T}$ is a spanning tree in $\widetilde{G}_i$, there exists an edge say $g = uw$ of $\widetilde{T}$ that belongs to the cut $\delta_{\widetilde{G}_i}(U)$ with $u \in U$ and $w \in \widetilde{V} \backslash U$. Since $\widetilde{T}_U$ is a spanning tree in $\widetilde{G}_i(U)$, there must exist a node $u' \in U$ such that $g' = uu'$ is an edge of $\widetilde{T}_U$. As $\widetilde{G}_i$ is 2-node connected, there must exist a path P between u' and w not going through u. Let $\mathcal{C}$ be the cycle formed by g together with g' and P. We claim that there is at least one edge, say e', in $\mathcal{C} \backslash \widetilde{T}$ such that the path $P_{e'}$ of $\mathcal{C}_{e'}$ contains both edges g and g'. In fact, if not, then the edge set $T^* = \{g, g'\} \cup \{\bigcup_{e \in \mathcal{C} \backslash \widetilde{T}} P_e\} \cup (\widetilde{T} \cap \mathcal{C})$ contains a cycle. As $T^* \subset \widetilde{T}$, this is impossible. In consequence, cycle $\mathcal{C}_{e'}$ contains g and g'. By (10), we have $b(e) = \rho'$ for some $\rho' \in I\!R$ and for all edge $e \in E_{\widetilde{G}_i}(V_{e'})$. As g and g' belong to $\mathcal{C}_{e'}$, from (8) we have $\rho = \rho'$, and therefore $b(e) = \rho$ for all $e \in E_{\widetilde{G}_i}(U \cup V_{e'})$. As $w \in V_{e'}$, $U' = U \cup V_{e'}$ strictly contains U. Moreover we have that $b(e) = \rho$ for all $e \in E_{\widetilde{G}_i}(U')$, which completes the proof of (7).

If e is an edge of $\delta_G(V_1, \ldots, V_p) \cap F_i$, we can show as above that $\widetilde{T}_0' = \widetilde{T}_0 \backslash \{e\}$ is a solution of $\mathcal{S}(G, \mathcal{F})$. As $ax^{\widetilde{T}_0'} = ax^{\widetilde{T}_0} = \alpha$ and thus $bx^{\widetilde{T}_0'} = bx^{\widetilde{T}_0} = \beta$, this implies that $b(e) = 0$. Also if f is an edge of $E_G(V_i)$, as by b) $G(V_i)$ is 2-$\mathcal{F}$-connected, $\widetilde{T}_0 \backslash \{f\} \in \mathcal{S}(G, \mathcal{F})$, and similary we get $b(f) = 0$.

All together we have

$$b(e) = \begin{cases} \rho & \text{for all } e \in \delta_{G_i}(V_1, \ldots, V_p), \\ 0 & \text{for all } e \in \cup(\cup_{j=1,\ldots,p} E_G(V_j)). \end{cases}$$

Thus $b = \rho a$. Since the face induced by $bx \geq \beta$ is a facet of MSIPND$(G, \mathcal{F})$, $b \geq 0$. Therefore $\rho > 0$ which ends the proof of the theorem. □

Remark that the partition inequalities have only coefficients 0 and 1. Our second class of inequalities, given in the following, may have non 0-1 coefficients.

4.2. Cut-cycle inequalities

In this section we introduce a further class of valid inequalities for the MSIPND$(G, \mathcal{F})$. These inequalities are induced by edge subsets of cuts, and may have arbitrary high coefficients.

Theorem 4.6. *Let $W \subset V$ and $T_1 = \{e_1, \ldots, e_s\}$, $s \geq 3$, be an edge subset of $\delta_G(W)$. Let $1 \leq q < s$ be an integer. Suppose that for every $i = 1, \ldots, s$, there is $j_i \in \{1, \ldots t\}$ such that $F_{j_i} \cap T_1 = \{e_i, \ldots, e_{i+q-1}\}$ (the indices are modulo s). Let $T_2 = \delta_G(W) \backslash (T_1 \cup (\bigcap_{i=1,\ldots,s} F_{j_i}))$. For $e \in \delta_G(W)$, let $r_e = |\{i \in \{1, \ldots, s\} \mid e \in \delta_G(W) \backslash F_{j_i}\}|$, and r be the smallest integer such that $r(s-q) \geq \max_{e \in T_2}\{r_e\}$. Then the inequality*

$$x(T_1) + rx(T_2) \geq \left\lceil \frac{s}{s-q} \right\rceil \tag{11}$$

is valid for MSIPND$(G, \mathcal{F})$.

Proof: The following inequalities are valid for MSIPND$(G, \mathcal{F})$.

$$x(\delta_{G_{j_i}}(W)) \geq 1 \qquad \text{for all } i = 1, \ldots, s, \tag{12}$$

$$(r(s-q) - r_e)x(e) \geq 0 \qquad \text{for all } e \in T_2. \tag{13}$$

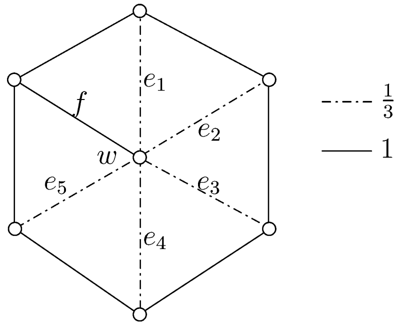
By summing these inequalities we obtain

$$(s-q)x(T_1) + r(s-q)x(T_2) \geq s.$$

By dividing this inequality by $s - q$ and rounding up the right hand side we get inequality (11). □

Inequalities (11) will be called *cut-cycle inequalities*.

Fig. 3 A fractional extreme point



Consider for example the graph $G = (V, E)$ shown in Fig. 3. Let $\mathcal{F} = \{F_1, \ldots, F_5\}$ such that $F_i = \{f, e_i, e_{i+1}\}$ (the indices are modulo 5). Let $W = \{w\}$. Let $T_1 = \{e_1, \ldots, e_5\}$. Observe that $F_i \subset \delta(W)$ for $i = 1, \ldots, 5$ and $T_2 = \emptyset$. By Theorem 4.6, it follows that the inequality $x(T_1) \geq 2$ is valid for MSIPND$(G, \mathcal{F})$. Moreover we can see that this cut-cycle inequality cuts the extreme point given by $\bar{x}(e_i) = \frac{1}{3}$ for $i = 1, \ldots, 5$, $\bar{x}(f) = 1$ and $\bar{x}(e) = 1$ otherwise. The solution $\bar{x}$ is an extreme point of the polytope given by the cut and the partition inequalities.

In what follows we restrict ourselves to the case where $q = 1$ and $\delta_G(W)\backslash(T_1 \cup T_2) = \emptyset$. We give necessary and sufficient conditions for inequality (11) to be facet defining in this case. When solving the MSIPND$(G, \mathcal{F})$ by cutting planes, we remarked that most of cut-cycle violated inequalities are of that type.

Theorem 4.7. *Suppose $q = 1$ and $\delta_G(W)\backslash(T_1 \cup T_2) = \emptyset$. Then inequality* (11) *defines a facet of MSIPND$(G, \mathcal{F})$ if and only if the following conditions are satisfied.*

(a) *T_1 is maximal, that is for every edge e of T_2 there is $i \in \{1, \ldots, s\}$ such that $e \in F_{j_i}$. (Note that in this case we have $r = 1$.)*

(b) *$G(W)$ and $G(\overline{W})$ are $\mathcal{F}$-connected.*

(c) *If $G(W)$ (resp. $G(\overline{W})$) is not 2-$\mathcal{F}$-connected, then for every edge $e \in E_G(W)$ such that $G(W)\backslash e$ (resp. $G(\overline{W})\backslash e$) is not $\mathcal{F}$-connected, there are two edges $g_1, g_2 \in \delta_G(W)$ such that*

(c.1) *$g_1, g_2 \notin F_i$, for all $i \in I_e$,*

(c.2) *$|F_j \cap \{g_1, g_2\}| \leq 1$ for all $j \in \{1, \ldots, t\}\backslash I_e$,*

(c.3) *for every subset W_1 of W (resp. W_1' of $\overline{W}$) such that $\delta_G(W_1, W\backslash W_1) \subset F_i \cup \{e\}$ (resp. $\delta_G(W_1', \overline{W}\backslash W_1') \subset F_i \cup \{e\}$) for some $i \in I_e$, we have $|\delta_G(W_1, \overline{W}) \cap \{g_1, g_2\}| = 1$ (resp. $|\delta_G(W_1', \overline{W}) \cap \{g_1, g_2\}| = 1$).*

Here I_e denotes the set $\{i \in \{1, \ldots, t\} \mid G_i(W)\backslash e$ is not connected$\}$ (resp. $\{i \in \{1, \ldots, t\} \mid G_i(\overline{W})\backslash e$ is not connected$\}$.

Proof: *Necessity*:

(a) Suppose there is an edge f of T_2 that does not belong to any of the sets F_{j_i}, $i = 1, \ldots, s$. Then $r_f = s$. As $q = 1$, it follows that $r = 2$ and therefore the cut-cycle inequality corresponding to T_1 and T_2, in this case, can be written as

$$x(T_1) + 2x(T_2) \geq 2. \tag{14}$$

Since every edge of E belongs to some F_i, there is $l \in \{1, \ldots, t\} \backslash \{j_1, \ldots, j_s\}$ such that $f \in F_l$. Consider the sets $T_1' = T_1 \cup \{f\}$ and $T_2' = T_2 \backslash \{f\}$. Consider the valid inequalities

$$\begin{array}{ll} x(\delta_{G_l}(W)) \geq 1, & \\ x(e) \geq 0 & \text{for all } e \in (T_1 \cap F_l), \\ 2x(e) \geq 0 & \text{for all } e \in (T_2 \cap F_l) \backslash \{f\}, \\ x(f) \geq 0. & \end{array}$$

By summing these inequalities, the inequalities (12) and the inequalities (13) for all $e \in T_2 \backslash \{f\}$, we obtain

$$sx(T_1') + 2sx(T_2') \geq s + 1.$$

(Recall that $r = 2$.) Dividing by s and rouding up the right hand side, imply that the inequality

$$x(T_1') + 2x(T_2') \geq 2$$

is valid for MSIPND($G, \mathcal{F}$). As this inequality dominates (14), the latter one can not define a facet.

Thus T_1 must be maximal. And as a consequence, we have $r_e \leq s - 1$ for all $e \in T_2$. As $q = 1$, it then follows that $r = 1$.

(b) Suppose that for some $i \in \{1, \ldots, t\}$, we have for instance that, $G_i(W)$ is not connected. Then there is a partition W_1, W_2 of W such that $\delta_{G_i}(W_1, W_2) = \emptyset$. Hence $\delta_{G_i}(W_1) = \delta_{G_i}(W_1, \overline{W})$ and $\delta_{G_i}(W_2) = \delta_{G_i}(W_2, \overline{W})$. Thus

$$\begin{aligned} x(\delta_{G_i}(W)) &= x(\delta_{G_i}(W_1, \overline{W})) + x(\delta_{G_i}(W_2, \overline{W})) \\ &= x(\delta_{G_i}(W_1)) + x(\delta_{G_i}(W_2)) \\ &\geq 2. \end{aligned}$$

As $\delta_{G_i}(W) \subseteq T_1 \cup T_2$ and $x(e) \geq 0$ for all $e \in E$, we have, in consequence, $x(T_1) + x(T_2) \geq 2$. But this inequality dominates(14), and hence (14) is not facet defining.

(c) We will prove the statement for $G(W)$, the proof is similar for $G(\overline{W})$. So suppose that $G(W)$ is not 2-$\mathcal{F}$-connected. Suppose that the statement does not hold for an edge $e \in E_G(W)$. We claim that every solution T of $\mathcal{S}(G, \mathcal{F})$ such that $|T \cap \delta_G(W)| = 2$ contains e. In fact, assume on the contrary, that $e \notin T$. Let g_1, g_2 be the edges of $\delta_G(W)$ in T. Thus g_1, g_2 do not satisfy at least one of the conditions (c.1), (c.2), (c.3).

- If g_1, g_2 do not satisfy (c.1), then there is $i \in I_e$ such that for instance $g_1 \in F_i$. Then there is $W_1 \subset W$ such that $\delta_G(W_1, W \backslash W_1) \cap T \subseteq F_i \cup \{e\}$. W.l.o.g., we may assume that $g_1 \in \delta_G(W_1, \overline{W})$. If $g_2 \in \delta_G(W \backslash W_1, \overline{W})$ (resp. $g_2 \in \delta_G(W_1, \overline{W})$), then $\delta_{G_i}(W_1) \cap T = \emptyset$ (resp. $\delta_{G_i}(W \backslash W_1) \cap T = \emptyset$). Hence the graph induced by $T \backslash F_i$ is not connected, a contradiction.
- If there is $j \in \{1, \ldots, t\} \backslash I_e$ such that $g_1, g_2 \in F_j$ then clearly $T \backslash F_j$ does not induce a connected graph, a contradiction.
- Now suppose that g_1, g_2 do not satisfy (c.3). Then we may assume that there is a partition $(W_1, W \backslash W_1)$ of W and $i \in I_e$ such that $\delta_G(W_1, W \backslash W_1) \subset F_i \cup \{e\}$ and $\delta_G(W_1, \overline{W})$ either contains $\{g_1, g_2\}$ or does not intersect this set. If $\{g_1, g_2\} \subset \delta_G(W_1, \overline{W})$ (resp. $\{g_1, g_2\} \cap \delta_G(W_1, \overline{W}) = \emptyset$), then $\delta_{G_i}(W_1) \cap T = \emptyset$ (resp. $\delta_{G_i}(W \backslash W_1) \cap T = \emptyset$).

Hence $T \notin \mathcal{S}(G, \mathcal{F})$, a contradiction.

Thus for all $T \in \mathcal{S}(G, \mathcal{F})$ such that $|T \cap \delta_G(W)| = 2$, we have $e \in T$. This implies that the face defining by (14) is contained in $\{x \in \text{ MSIPND}(G, \mathcal{F})\ \}\ x(e) = 1\}$. As (14) is not a trivial inequality, it follows that (14) does not define a facet.

Sufficiency: Suppose $\delta_G(W) \cap (T_1 \cup T_2) = \emptyset$ and $q = 1$. Suppose also that conditions (a), (b) and (c) are satisfied. By (a) inequality (11) can be written as

$$x(T_1) + x(T_2) \geq 2.$$

Let us denote inequality (11) by $ax \geq \alpha$, and let $bx \geq \beta$ be a facet defining inequality of MSIPND$(G, \mathcal{F})$ such that $\{x \in \text{ MSIPND}(G, \mathcal{F})\ \}\ ax = \alpha\} \subseteq \{x \in \text{MSIPND}(G, \mathcal{F})\ \}\ bx = \beta\}$. To show that $ax \geq \alpha$ defines a facet we will show as before that there is $\rho > 0$ such that $b = \rho a$.

As $s \geq 3$, let e_1, e_2 be two edges of T_1. Let

$$\Gamma_0 = \{e_1, e_2\} \cup E_G(W) \cup E_G(\overline{W}).$$

As e_1 and e_2 belong to different F_i's, and by condition (b) both graph $G(W)$ and $G(\overline{W})$ are $\mathcal{F}$-connected, it follows that $\Gamma_0 \in \mathcal{S}(G, \mathcal{F})$. Consider the edge sets $\Gamma_i = (\Gamma_0 \backslash \{e_1\}) \cup \{e_i\}$ for $i = 3, \ldots, s$. Clearly, $\Gamma_i \in \mathcal{S}(G, \mathcal{F})$ for $i = 3, \ldots, s$. Moreover we have $ax^{\Gamma_0} = ax^{\Gamma_i} = \alpha$. Hence $bx^{\Gamma_i} = bx^{\Gamma_0} = \beta$ and in consequence $b(e_i) = b(e_1)$ for $i = 3, \ldots, s$. By symmetry, we then obtain that

$$b(e) = \rho \text{ for some } \rho \in I\!R, \text{ for all } e \in T_1. \tag{15}$$

Now consider an edge $f \in T_2$. By condition (a), there is $i \in \{1, \ldots, s\}$ such that $f \in F_{j_i}$. Consider the set $\Gamma_f = (\Gamma_i \backslash \{e_i\}) \cup \{f\}$ (resp. $\Gamma_f = (\Gamma_0 \backslash \{e_i\}) \cup \{f\}$) if $i \in \{3, \ldots, s\}$ (resp. $i \in \{1, 2\}$) where Γ_0 and Γ_i are the edge sets introduced above. Clearly $ax^{\Gamma_f} = ax^{\Gamma_i} = ax^{\Gamma_0} = \alpha$. Thus $bx^{\Gamma_f} = bx^{\Gamma_i} = bx^{\Gamma_0} = \beta$, and therefore $b(f) = b(e_i)$. This together with (15) yield

$$b(e) = \rho \qquad \text{for all } e \in T_1 \cup T_2. \tag{16}$$

Consider an edge $e \in E_G(W)$. If $G(W) \backslash e$ is $\mathcal{F}$-connected, then $\Gamma_0 \backslash \{e\}$ is a solution of $\mathcal{S}(G, \mathcal{F})$, and hence $b(e) = 0$. If not, then by (c), there are two edges g_1, g_2 that satisfy conditions (c.1), (c.2), (c.3). Let $\Gamma'_0 = (\Gamma_0 \backslash \{e_1, e_2, e\}) \cup \{g_1, g_2\}$. We claim that $\Gamma'_0 \in \mathcal{S}(G, \mathcal{F})$. In the following, we will denote by H_i the subgraph of G induced by $\Gamma'_0 \backslash F_i$, for $i \in \{1, \ldots, t\}$. Let F_i such that $i \in I_e$ (where I_e is defined in condition (c)). By (c.1), $g_1, g_2 \notin F_i$. As $G(\overline{W})$ is $\mathcal{F}$-connected, if H_i is not connected, then there must exist $W_1 \subset W$, such that $\delta_{H_i}(W_1) = \emptyset$. Since $E_G(W) \backslash \{e\} \subset \Gamma'_0$, it follows that $\delta_G(W_1, W \backslash W_1) \subset F_i \cup \{e\}$ and $g_1, g_2 \in \delta_G(W \backslash W_1, \overline{W})$. But this contradict (c.3).

Now consider a set F_i with $i \in \{1, \ldots, t\} \backslash I_e$. Then $G_i(W) \backslash e$ is connected. As by (c.2), $\{g_1, g_2\} \backslash F_i \neq \emptyset$, and $G(\overline{W})$ is $\mathcal{F}$-connected, we have that H_i is connected. Consequently $\Gamma'_0 \in \mathcal{S}(G, \mathcal{F})$. Also we have $ax^{\Gamma'_0} = \alpha$. Hence $bx^{\Gamma'_0} = \beta$. As $bx^{\Gamma_0} = \beta$ and by (16), $b(g_1) = b(g_2) = b(e_1) = b(e_2)$, we obtain that $b(e) = 0$.

Thus $b(e) = 0$ for all $e \in E_G(W)$. Similarly we have that $b(e) = 0$ for all $e \in E_G(\overline{W})$. This together with (16) yield $b = \rho a$. Moreover we should also have $\rho > 0$, and the proof is complete. □

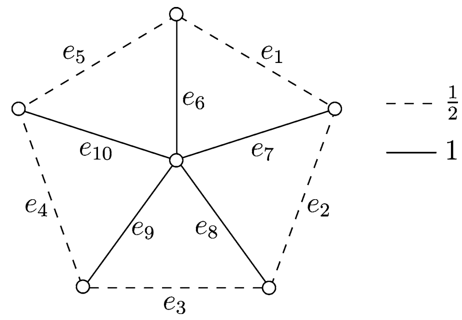


Fig. 4 A fractional extreme point

Consider the graph displayed in Fig. 4 and the edge sets $F_i = \{e_{i+5}\}$ for $i = 1, \ldots, 5$. Let $\bar{x}$ be the solution given by $\bar{x}(e_i) = \frac{1}{2}$ for $i = 1, \ldots, 5$ and $\bar{x}(e_i) = 1$ for $i = 6, \ldots, 10$. Clearly, $\bar{x}$ satisfies the trivial and the cut inequalities with respect to $F_1, \ldots, F_5$. Solution $\bar{x}$ also satisfies the partition and the cut-cycle inequalities. Moreover, it is not hard to see that $\bar{x}$ is an extreme point of the polytope given by these inequalities. However, $\bar{x}$ violates the inequality

$$x(e_1) + x(e_2) + x(e_3) + x(e_4) + x(e_5) \geq 3,$$

which is valid for the associated polytope. In what follows we will show this as a special case of a more general class of valid inequalities for the MSIPND(G, $\mathcal{F}$).

4.3. star-partition inequalities

Let $G = (V, E)$ be a graph and $\mathcal{F} = \{F_1, \ldots, F_t\}$, with $t \geq 2$, be a family of edge subsets of E. Let $V_0, V_1, \ldots, V_p$ be a partition of V with p odd. Suppose that for every $i = 1, \ldots, p$, there is $j_i \in \{1, \ldots, t\}$ such that $F_{j_i} \cap [V_i, V_0] \neq \emptyset$. Let $\Lambda = \{e \in E \mid e \in [V_k, V_l] \cap F_{j_k} \cap F_{j_l}$, for some $k, l \in \{1, \ldots, p\}\}$. Let $F = \bigcup_{i=1}^{p} (F_{j_i} \cap [V_i, V_0]) \cup \Lambda$. Such a configuration will be called a *star-partition configuration* (see Fig. 5).

Consider the inequality

$$x(\delta_G(V_0, \ldots, V_p) \backslash F) \geq \left\lceil \frac{p}{2} \right\rceil. \tag{17}$$

Theorem 4.8. *Inequality* (17) *is valid for the MSIPND(G, $\mathcal{F}$).*

Proof: Clearly, the following inequalities are valid for the MSIPND(G, $\mathcal{F}$).

$$\begin{aligned}
x(\delta_{G \backslash F_{j_i}}(V_i)) &\geq 1 && \text{for all } i = 1, \ldots, p, \\
x(e) &\geq 0 && \text{for all } e \in \delta(V_0) \backslash F, \\
x(e) &\geq 0 && \text{for all } e \in ([V_k, V_l] \cap F_{j_k}) \backslash F_{j_l},\ k = 1, \ldots, p,\ l = 1, \ldots, p,\ k \neq l.
\end{aligned}$$

By summing these inequalities, we obtain the inequality

$$2x(\delta(V_0, \ldots, V_p) \backslash F) \geq p.$$

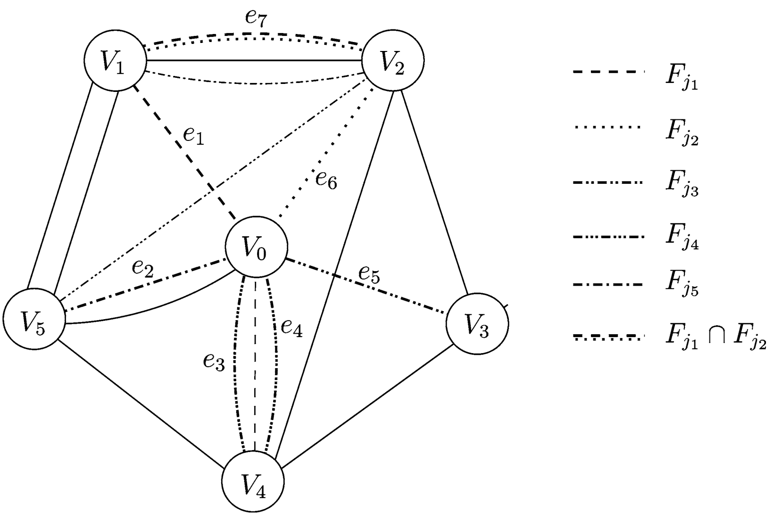


Fig. 5 A star-partition configuration

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

For example, consider the star-partition configuration given in Fig. 5. The corresponding partition has six elements $V_0, V_1, \ldots, V_5$. With each V_i it is associated an edge set F_{j_i} such that $F_{j_i} \cap [V_i, V_0] \neq \emptyset$ for $i = 1, \ldots, 5$. These sets are represented by different types of lines. Note that edge e_7 belongs to both sets F_{j_1} and F_{j_2}, and hence $\Lambda = \{e_7\}$. The star-partition inequality corresponding to this configuration is given by

$$x(\delta(V_0, V_1, \ldots, V_5)\backslash F) \geq 3,$$

where $F = \{e_1, \ldots, e_6, e_7\}$.

Inequalities (17) will be called *star-partition inequalities*.

Inequalities similar to (17) called F-partition inequalities have been introduced in Mahjoub (1994) for the 2-edge (node) connected subgraph problem.

Let $R_{j_i} = F_{j_i} \cap [V_i, V_0]$ for $i = 1, \ldots, p$ and $\Delta = F \cup (\bigcup_{i=0,\ldots,p} E_G(V_i))$.

Theorem 4.9. *Inequality* (17) *defines a facet of MSIPND($G, \mathcal{F}$) only if the following hold.*

(a) *There is an edge set* $\widetilde{T} \subseteq \delta_G(V_1, \ldots, V_p)$ *such that* $\widetilde{T} \cup \Delta \in \mathcal{S}(G, \mathcal{F})$ *and* $|\widetilde{T}| = \frac{p+1}{2}$.

(b) *For every edge* $e \in [V_i, V_{i'}]$, $i, i' \in \{1, \ldots, p\}$, *there is no* $i_0 \in \{1, \ldots, t\}$ *such that* $(R_{j_i} \cup R_{j_{i'}} \cup \{e\}) \subseteq F_{i_0}$.

Proof: Suppose (17) defines a facet of MSIPND($G, \mathcal{F}$). It is clear that there exists a solution $T \in \mathcal{S}(G, \mathcal{F})$ such that $|T \cap (\delta_G(V_0, \ldots, V_p)\backslash F)| = \frac{p+1}{2}$. For otherwise the face defined by (17) would be empty, and hence can not be a facet. Let $T' = T \cap (\delta_G(V_0, \ldots, V_p)\backslash F)$. Suppose that $T' \cap \delta_G(V_0) \neq \emptyset$. We claim that $|T' \cap \delta_G(V_0)| = 1$. In fact, suppose that this is not the case and let f_1, f_2 be two edges of $T' \cap \delta_G(V_0)$. Let $i_1, i_2 \in \{1, \ldots, p\}$ such that $f_1 \in [V_{i_1}, V_0]$ and $f_2 \in [V_{i_2}, V_0]$, and suppose, w.l.o.g., that $i_1 \neq i_2$. As $T \in \mathcal{S}(G, \mathcal{F})$, and hence $G[T\backslash F_{j_i}]$ is connected for $i = 1, \ldots, p$, one should have $T \cap (\delta_G(V_i)\backslash F_{j_i}) \neq$

$\emptyset$ for all $i \in \{1, \ldots, p\} \backslash \{i_1, i_2\}$. As $T \cap (\delta_G(V_i) \backslash F_{j_i}) = T' \cap (\delta_G(V_i) \backslash F_{j_i})$, we then have $T' \cap (\delta_G(V_i) \backslash F_{j_i}) \neq \emptyset$ for all $i \in \{1, \ldots, p\} \backslash \{i_1, i_2\}$. Since p is odd, this implies that $|T'| \geq \frac{p+3}{2}$, a contradiction.

In consequence, any solution of $\mathcal{S}(G, \mathcal{F})$, whose incidence vector satisfies (17) with equality, contains at most one edge of $\delta_G(V_0) \backslash F$. If a) does not hold, then every solution of $\mathcal{S}(G, \mathcal{F})$ whose incidence vector satisfies (17) with equality contains exactly one edge of $\delta_G(V_0) \backslash F$. But this implies that the face defined by (17) is contained in the hyperplane $\sum_{e \in \delta_G(V_0) \backslash F} x(e) = 1$.But, this contradicts the fact that (17) defines a facet.

Now consider an edge $e \in [V_i, V_{i'}]$ for some $i, i' \in \{1, \ldots, p\}$. Since (17) is different from inequality $x(e) \geq 0$, there must exist a solution, say T_1, of $\mathcal{S}(G, \mathcal{F})$ that contains e and whose incidence vector satisfies (17) with equality. Let $\widetilde{T}_1 = T_1 \backslash \Delta$. Since $|\widetilde{T}_1| = \frac{p+1}{2}$, e can be adjacent to at most one edge of $\widetilde{T}_1 \backslash \{e\}$ and $[V_i, V_{i'}] \cap \widetilde{T}_1 = \{e\}$. If $R_{j_i} \cup R_{j_{i'}} \cup \{e\} \subseteq F_{i_0}$ for some $i_0 \in \{1, \ldots, t\}$, then $G[T_1 \backslash F_{i_0}]$ is not connected, which is impossible. □

Theorem 4.10. *Inequality* (17) *defines a facet of MSIPND($G, \mathcal{F}$) if*

(a) *condition* (a) *of Theorem* (4.9) *is satisfied,*

(b) $G(V_i)$ *is 2-$\mathcal{F}$-connected for* $i = 0, 1, \ldots, p$,

(c) *for all* $i, i' \in \{1, \ldots, p\}$, $e \in R_{j_i}$ *and* $l \in \{1, \ldots, t\}$, $(R_{j_i} \cup R_{j_{i'}}) \backslash (F_l \cup \{e\}) \neq \emptyset$,

(d) *for all* $e \in [V_i, V_{i'}]$, $i, i' \in \{1, \ldots, p\}$, *there is no* $i_0 \in \{1, \ldots, t\}$ *with either* $F_{j_i} \cup \{e\} \subseteq F_{i_0}$ *or* $F_{j_{i'}} \cup \{e\} \subseteq F_{i_0}$.

Proof: Let denote (17) by $ax \geq \alpha$ and let $bx \geq \beta$ be a facet defining inequality of MSIPND($G, \mathcal{F}$) such that $\{x \in$ MSIPND$(G, \mathcal{F}) \mid ax = \alpha\} \subseteq \{x \in$ MSIPND$(G, \mathcal{F}) \mid bx = \beta\}$. As we did before, we will show that $b = \rho a$ for some $\rho > 0$.

Let $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ be the graph obtained by contracting the sets $V_0, \ldots, V_p$. Let $p = 2k + 1$, $k \geq 1$. By condition a) of Theorem 4.9 there is an edge set $\widetilde{T} \subseteq \delta_G(V_1, \ldots, V_p)$ such that $|\widetilde{T}| = k + 1$ and $T_0 = \widetilde{T} \cup \Delta \in \mathcal{S}(G, \mathcal{F})$. Hence $\widetilde{T} \cap \delta_G(V_i) \neq \emptyset$ for $i = 1, \ldots, 2k + 1$. After eventual permutation of the sets $V_1, \ldots, V_{2k+1}$, one may suppose that $\widetilde{T} = \{e_1, e_3, \ldots, e_{k+1}\}$ such that $e_i \in [V_{2i-1}, V_{2i}]$ for $i = 1, 2, \ldots, k + 1$ (where the indices are taken modulo $2k + 1$). Observe that $\widetilde{T} \cap \delta(V_1) = \{e_1, e_{k+1}\}$ and the edges of $\widetilde{T} \backslash \{e_1\}$ are pairwise non-adjacent. Let e be an edge of $\delta_G(V_2) \backslash F_{j_2}$ between V_2 and V_l, $l \neq 2$. Consider the edge set $T_e = (T_0 \backslash \{e_1\}) \cup \{e\}$. We claim that $T_e \in \mathcal{S}(G, \mathcal{F})$.
Since by (b) $G(V_i)$ is 2-$\mathcal{F}$-connected and hence $\mathcal{F}$-connected, for $i = 0, 1, \ldots, 2k + 1$, it suffices to show that $\widetilde{G}[T_e \backslash F_j]$ is connected for $j = 1, \ldots, t$.

Consider a node V_i, $i \in \{1, \ldots, p\}$. Let f be the edge of $T_e \backslash \Delta$ incident to V_i. Note that $f \in (\widetilde{T} \backslash \{e_1\}) \cup \{e\}$. Let i' such that $f \in [V_i, V_{i'}]$. By (c), we have $(R_{j_i} \cup R_{j_{i'}}) \backslash F_j \neq \emptyset$. And by (d) we have that $(F_{j_i} \cup \{f\}) \backslash F_j \neq \emptyset \neq (F_{j_{i'}} \cup \{f\}) \backslash F_j$. Hence in $\widetilde{G}[T_e \backslash F_j]$, V_i is linked to V_0 by a path consisting of either one or two edges.

In consequence, in $\widetilde{G}[T_e \backslash F_j]$ all the nodes V_i, $i = 1, \ldots, p$ are linked to V_0. Therefore $\widetilde{G}[T_e \backslash F_j]$ is connected, yielding T_e is a solution of $\mathcal{S}(G, \mathcal{F})$.
Since $ax^{T_0} = ax^{T_e}$, we have $bx^{T_0} = bx^{T_e}$ and hence $b(e_1) = b(e)$. Since e is an arbitrary edge of $\delta_G(V_2) \backslash F_{j_2}$ it follows that there is ρ_2 such that

$$b(e) = \rho_2 \quad \text{for all } e \in \delta_G(V_2) \backslash F_{j_2}. \tag{18}$$

Consider a node V_i, $i \in \{1, \ldots, p\}\backslash\{2\}$, and let $i' \in \{1, \ldots, p\}\backslash\{i\}$ such that $[V_i, V_{i'}] \cap (\widetilde{T}\backslash\{e_1\}) \neq \emptyset$. Note that either $i' = i + 1$ or $i' = i - 1$. Since G is complete, $[V_{i'}, V_2] \neq \emptyset$. Let g be an edge of $[V_{i'}, V_2]$, and consider the edge set $T_g = (T_0\backslash\{e_1\}) \cup \{g\}$. By the above development, $T_g \in \mathcal{S}(G, \mathcal{F})$. Also note that T_g has a structure similar to that of T_0, that is $|T_g\backslash\Delta| = k + 1$. Now along the same line as above, we can show that there is ρ_i such that

$$b(e) = \rho_i \quad \text{for all } e \in \delta_G(V_i)\backslash F_{j_i}. \tag{19}$$

From (18) and (19), it follows that $\rho_i = \rho$ for $i = 1, \ldots, p$ and some $\rho \in \mathbb{R}$, and therefore $b(e) = \rho$ for all $e \in \delta_G(V_0, \ldots, V_p)\backslash F$.
If e is an edge of $E_G(V_i)$, for $i \in \{0, \ldots, p\}$, as $G(V_i)$ is 2-$\mathcal{F}$-connected, $T_0' = T_0\backslash\{e\}$ is a solution of $\mathcal{S}(G, \mathcal{F})$ and hence $b(e) = 0$.

Now let e be an edge of R_{j_i} for $i \in \{1, \ldots, p\}$. We claim that $\overline{T}_0 = T_0\backslash\{e\} \in \mathcal{S}(G, \mathcal{F})$. Let $i' \in \{1, \ldots, p\}\backslash\{i\}$ such that $[V_i, V_{i'}] \cap \widetilde{T} \neq \emptyset$. Let $j \in \{1, \ldots, t\}$. By (c) we have $(R_{j_i} \cup R_{j_{i'}})\backslash F_j \neq \emptyset$. If $e \in F_j$, then $G(\overline{T}_0\backslash F_j) = G(T_0\backslash F_j)$, and therefore $G(\overline{T}_0\backslash F_j)$ is connected. So suppose $e \notin F_j$. By (c) $(R_{j_i} \cup R_{j_{i'}})\backslash(F_j \cup \{e\}) \neq \emptyset$. Then in $\widetilde{G}[\widetilde{T}_0\backslash F_j]$ both nodes V_i and $V_{i'}$ are linked to V_0. As $T_0 \in \mathcal{S}(G, \mathcal{F})$ and hence all the other nodes are linked to V_0, we have that $\widetilde{G}[\overline{T}_0\backslash F_j]$ is connected. Consequently, we get $b(e) = 0$. Similary we show that $b(e) = 0$ for all $e \in \Lambda$. We then have shown that

$$b(e) = \begin{cases} \rho & \text{for all } e \in \delta_G(V_1, \ldots, V_p)\backslash F, \\ 0 & \text{for all } e \in \Delta. \end{cases}$$

Thus $b = \rho a$. As $bx \geq \beta$ is a facet defining, we also have $\rho > 0$. $\square$

5. Reduction operations

Let $\mathcal{P}(G, \mathcal{F})$ be the polytope given by inequalities (4) and (5). In this section we introduce some reduction operations defined with respect to a solution x of $\mathcal{P}(G, \mathcal{F})$. These permit to reduce G and $\mathcal{F}$ to G' and $\mathcal{F}'$ and x to x' so that if x is an extreme point of $\mathcal{P}(G, \mathcal{F})$, then x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$. As it will turn out, these operations are useful in a preprocessing phase of a cutting plane based algorithm for the MSIPND problem, then allow to much accelerate the resolution of the problem. These operations use some ideas similar to those developed by Kerivin et al. (Kerivin, Mahjoub, and Nocq, 2004) for the 2-edge connected subgraph problem.

Given a solution x of $\mathcal{P}(G, \mathcal{F})$, we denote by $E_0(x)$ (resp. $E_1(x)$) the set of edges $e \in E$ such that $x(e) = 0$ (resp. $x(e) = 1$). We also denote by $\tau_i(x)$ the set of cuts $\delta_{G_i}(W)$ tight for x, that is $x(\delta_{G_i}(W)) = 1$.

If x is an extreme point of $\mathcal{P}(G, \mathcal{F})$, then there are subsets $\tau_i'(x) \subseteq \tau_i(x)$, $i = 1, \ldots, t$ such that x is the unique solution of the system

$$S(x) \quad \begin{cases} x(e) = 0 & \text{for all} \quad e \in E_0(x), \\ x(e) = 1 & \text{for all} \quad e \in E_1(x), \\ x(\delta_{G_i}(W)) = 1 & \text{for all} \quad \delta_{G_i}(W) \in \tau_i'(x), \quad i = 1, \ldots, t, \end{cases}$$

where $|E_0(x)| + |E_1(x)| + \sum_{i=1,\ldots,t} |\tau_i'(x)| = |E|$.

Consider a solution x of $\mathcal{P}(G, \mathcal{F})$, we have the following lemmas.

Lemma 5.1. *Let $f \in E$ be an edge such that $x(f) = 0$. Let $G' = (V', E')$ be the graph obtained from G by deleting f. Let $\mathcal{F}' = \{F_1', \ldots, F_t'\}$ with $F_i' = F_i \backslash \{f\}$ for $i = 1, \ldots, t$. Let x' be the restriction of x on E'. Then x is an extreme point of $\mathcal{P}(G, \mathcal{F})$ if and only if x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$.*

Proof: Easy. □

Lemma 5.2. *Let $W \subseteq V$ be a node subset of V with $\delta_G(W) = \{f, g\}$ such that $x(f) = x(g) = 1$. Suppose that f belongs to only one set of $\mathcal{F}$, say F_{i_0}, and that $F_{i_0} = \{f\}$. Let $G' = (V', E')$ be the graph obtained from G by contracting f. Let $\mathcal{F}' = \{F_1', \ldots, F_t'\}$ with $F_i' = F_i \backslash \{f\}$ for $i = 1, \ldots, t$. If x is an extreme point of $\mathcal{P}(G, \mathcal{F})$ then x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$ where x' is the restriction of x on E'.*

Proof: Suppose x is an extreme point of $\mathcal{P}(G, \mathcal{F})$. First observe that as $\delta_G(W) = \{f, g\}$, f, g do not belong to the same F_i. For otherwise, one would have $x(\delta_{G_i}(W)) = 0 < 1$, contradicting the fact that x is a solution of $\mathcal{P}(G, \mathcal{F})$. We claim that the cuts of system $S(x)$ are also cuts in G' and, hence tight for x'. For this it suffices to show that for every cut $\delta_{G_i}(U)$ of $S(x)$, we have $\delta_{G_i}(U) = \delta_{G_j'}(U')$ for some $U' \subset V'$ and $j \in \{1, \ldots, t\}$, where G_j' is the graph obtained from G' by deleting F_j'. Indeed, first note that, since $x(f) = x(g) = 1$ and $S(x)$ is nonsingular, $f, g \notin \delta_{G_i}(U)$. If f belongs to $E_G(U)$ (resp. $f \in E_G(\overline{U})$), then clearly $\delta_{G_i}(U) = \delta_{G_i'}(U')$, where $U' = \overline{U}$ (resp. $U' = U$). Note that this case applies if $i \neq i_0$.

So suppose that $i = i_0$, and that $f \in \delta_G(U)$. Let $J \subset \{1, \ldots, t\}$ such that $g \in F_j$ for all $j \in J$. Let $U_1 = U \cap W$, $U_2 = W \backslash U$, $U_3 = \overline{W} \cap \overline{U}$, $U_4 = \overline{W} \cap U$. As $g \notin \delta_{G_{i_0}}(U)$, we may, w.l.o.g., assume that $g \in [U_2, U_3]$. Also we may suppose that $f \in [U_1, U_3]$ (see Fig. 6). Let $D = \delta_{G_{i_0}}(U)$. Note that, as $F_{i_0} = \{f\}$, $\delta_G(U) = D \cup \{f\}$. If $[U_3, U_4] \neq \emptyset$, then either $\delta_{G_{i_0}}(U_1)$ or $\delta_{G_{i_0}}(U_4)$ is strictly contained in $\delta_{G_{i_0}}(U)$. As $\delta_{G_{i_0}}(U)$ is tight for x, this implies that at least one of the cuts $\delta_{G_{i_0}}(U_1)$ and $\delta_{G_{i_0}}(U_4)$ is violated by x, a contradiction. Thus $[U_3, U_4] = \emptyset$, and, therefore $U_4 = \emptyset$. In consequence, we have $D = [U_1, U_2]$, and $\delta_G(U_2) = D \cup \{g\}$. Since $x(D) = x(\delta_{G_{i_0}}(U)) = 1$, it follows that any edge of D, that belongs to some F_j, $j \in J$, must have a zero value. For otherwise, we would have $x(\delta_{G_j}(U_2)) < 1$, which is impossible. Now let $U' = U_2$. We have that $\delta_{G_{i_0}}(U) = \delta_{G_j'}(U')$, and the claim is proved.

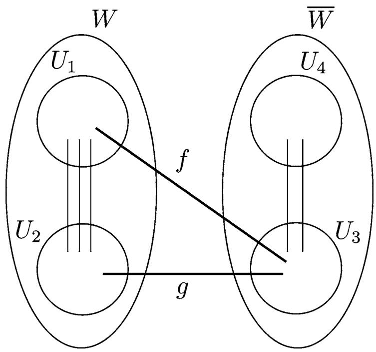


Fig. 6 A cut of cardinality two with one contractible edge

Let $S'(x)$ be the system obtained from $S(x)$ by deleting the equation $x(f) = 1$. In consequence x' is a solution of $S'(x)$. Since $S(x)$ is nonsingular, clearly $S'(x)$ is also nonsingular. As the equations of $S'(x)$ correspond to constraints of $\mathcal{P}(G', \mathcal{F}')$, we then have that x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$. $\square$

Lemma 5.3. *Let u, v be two nodes of V. Suppose there are two edges $f, g \in [u, v]$ such that*

(a) $x(f) = x(g) = 1$*, and*

(b) *there is no $i \in \{1, \ldots, t\}$ such that $f, g \in F_i$.*

Let $G' = (V', E')$ be the graph obtained by contracting $[u, v]$, and $\mathcal{F}' = \{F_1', \ldots, F_t'\}$ with $F_i' = F_i \backslash [u, v]$. Then, if x is an extreme point of $\mathcal{P}(G, \mathcal{F})$, x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$.

Proof: It is clear that $x' \in \mathcal{P}(G', \mathcal{F}')$. Let $\delta_{G_i}(U)$ be a cut of $\tau_i'(x)$. We claim that $\delta_{G_i}(U) \cap [u, v] = \emptyset$. Suppose not, then $[u, v] \backslash F_i \subset \delta_{G_i}(U)$. As by (b) f, g do not belong both to F_i, we may suppose, for instance, that $f \notin F_i$. Since $x(f) = 1$ and $x(g) > 0$, it then follows that $g \in F_i$ and $x(e) = 0$ for all $e \in [u, v] \backslash (F_i \cup \{f\})$. But this implies that $x(\delta_{G_i}(U)) = 1$ is redondant in $S(x)$, which contradicts the fact that $S(x)$ is nonsingular.
Thus, $\delta_G(U) \cap [u, v] = \emptyset$. In consequence, $\delta_{G_i'}(U) = \delta_{G_i}(U)$ and therefore $x'(\delta_{G_i'}(U)) = x(\delta_{G_i}(U)) = 1$. Let $\widetilde{S}(x)$ be the system obtained from $S(x)$ by deleting the equation $x(f) = 1$, $x(g) = 1$ and $x(e) = 0$ with $e \in [u, v]$. We then have that x' is a solution of $\widetilde{S}(x)$. Also note that all the equations of $\widetilde{S}(x)$ correspond to inequalities of $\mathcal{P}(G', \mathcal{F}')$. Since $S(x)$ is nonsingular, $\widetilde{S}(x)$ so is, which implies that x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$. $\square$

Lemma 5.4. *Let $W \subseteq V$ be a node set with $|W| \geq 2$ such that $G(W)$ is $\mathcal{F}$-connected and $x(e) = 1$ for all $e \in E_G(W)$. Let $G' = (V', E')$ be the graph obtained by contracting W. Let $\mathcal{F}' = \{F_1', \ldots, F_t'\}$ with $F_i' = F_i \backslash E_G(W)$. Let x' be the restriction of x on E'. If x is an extreme point of $\mathcal{P}(G, \mathcal{F})$, then x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$.*

Proof: Since $x' \in \mathcal{P}(G', \mathcal{F}')$, as we did before, we will show that x' is the unique solution of a subsystem of $S(x)$ that come from $\mathcal{P}(G', \mathcal{F}')$.
Let $\delta_{G_i}(U) \in \tau_i'(x)$ for some $U \subseteq V$ and $i \in \{1, \ldots, t\}$. We claim that either $W \subseteq U$ or $W \subseteq V \backslash U$. Indeed suppose, on the contrary, that $U \cap W \neq \emptyset \neq (V \backslash U) \cap W$. Let $u \in U \cap W$ and $u' \in (V \backslash U) \cap W$. Since $G(W)$ is $\mathcal{F}$-connected, we have that $G[E_G(W) \backslash F_i]$ is connected. Consequently, u and v must be linked by a path of $E_G(W) \backslash F_i$. Hence $\delta_{G_i}(U) \cap E_1(x) \neq \emptyset$. But this implies that $x(\delta_{G_i}(U)) = 1$ is redondant in $S(x)$, a contradiction. In consequence, all the cut equations of $S(x)$ correspond to cut inequalities of $\mathcal{P}(G', \mathcal{F}')$. Let $\bar{S}(x)$ be the system obtained from $S(x)$ by deleting the equations $x(e) = 1$, $e \in E_G(W)$. Hence x' is a solution of $\bar{S}(x)$. As $\bar{S}(x)$ is nonsingular, this implies that x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$. $\square$

Let $\theta_1, \ldots, \theta_4$ be the operations introduced in Lemmas 5.1– 5.4, respectively, that is

θ_1**:** Delete an edge e with $x(e) = 0$.
θ_2**:** Contract an edge f such that f belongs to a 2-edge cutset.
θ_3**:** Contract a set of parallel edges $[u, v]$ such that there are two edges $f, g \in [u, v]$ with $x(f) = x(g) = 1$ and f, g do not both belong to some F_i, $i \in \{1, \ldots, t\}$.
θ_4**:** Contract a node subset $W \subseteq V$ such that $|W| \geq 2$, $G(W)$ is $\mathcal{F}$-connected and $x(e) = 1$ for all $e \in E_G(W)$.

Operations $\theta_1, \ldots, \theta_4$ have interesting algorithmic consequences. In fact, note that these operations can be implemented in polynomial time and in any order. If x is an extreme point of $\mathcal{P}(G, \mathcal{F})$, by applying repeatedly operations $\theta_1, \ldots, \theta_4$, we get a graph $G' = (V', E')$, a set $\mathcal{F}'$ and a solution $x' \in I\!R^{E'}$ which is, by Lemmas 5.1–5.4, an extreme point of $\mathcal{P}(G', \mathcal{F}')$. As it will be shown in the next section, the separation of inequalities (4), (6), (11) and (17) with respect to x in G, reduces to the separation of theses inequalities in G' with respect to x'.

In Fig. 7(a) we display a fractional vector obtained by solving a MSIPND problem on graph G with 28 nodes and $|\mathcal{F}| = 46$. The dashed lines represent the edges with fractional values, which are all equal to 0.5 and the solid lines, the edges with value 1. By applying operations $\theta_1, \ldots, \theta_4$, we obtain the reduced graph G' of Fig. 7(b) as well as the corresponding fractional solution. The nodes u and u' of G' induce the violated cut-cycle inequalities

$$x(e_1) + x(e_2) + x(e_3) \geq 2,$$
$$x(e_4) + x(e_5) + x(e_6) \geq 2.$$

(The edges e_1, e_2, e_3 (resp. e_4, e_5, e_6) do not belong to the same F_i' of $\mathcal{F}'$). Note that the cut-cycle induced by u' does not appear as a node cut in G. By lifting these inequalities and adding them to the linear relaxation, we get the fractional solution displayed in Fig. 7(c). Denote by $\bar{G}$ the support graph of this solution. Using operations $\theta_1, \ldots, \theta_4$ we get the graph $\bar{G}'$ of Fig. 7(d). Let F be the set of edges incident to v_0 in $\bar{G}'$. Each edge of F belongs to a set $F_i' \in \mathcal{F}'$. By considering the nodes of $\bar{G}'$ as the elements $V_0, \ldots, V_3$ of a node partition of $\bar{G}'$ where $V_i = \{v_i\}$, for $i = 0, \ldots, 3$, we obtain the following star-partition inequality

$$x(\delta_{\bar{G}'}(V_0, \ldots, V_3)\backslash F) \geq 2.$$

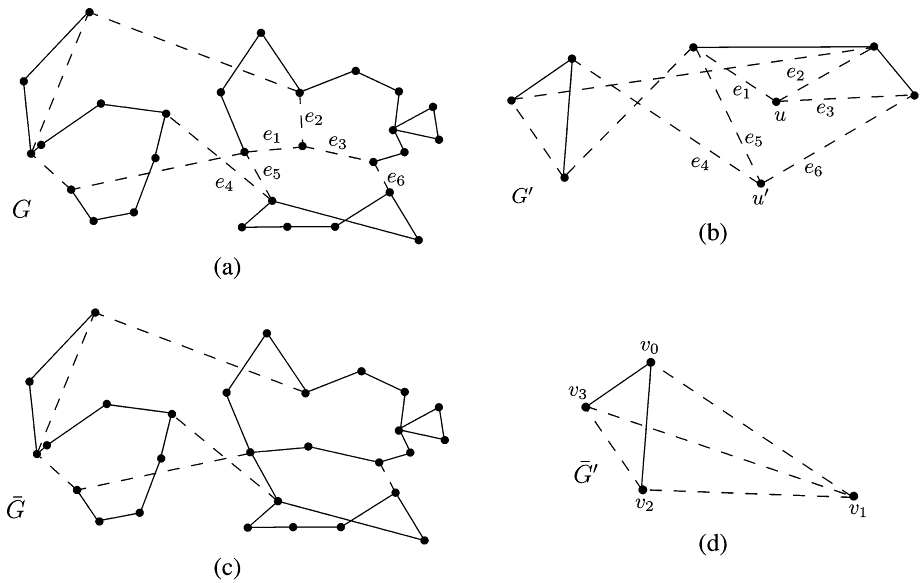


Fig. 7 Fractional solutions and reduced graphs

This inequality is violated by the solution of $\bar{G}'$. By lifting this inequality, we get a violated star-partition inequality that cuts off the fractional solution of $\bar{G}$.

6. A Branch-and-cut algorithm

In this section, we describe a Branch-and-Cut algorithm for the MSIPND problem. Our aim is to address the algorithmic applications of the results presented in the previous section. For this let us assume we are given two graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$ representing respectively the IP and the optical networks. Suppose also given a family $\mathcal{F} = \{F_e \subseteq E^1,\ e \in E^2\}$ of edge subsets of E^1 associated with the edges of G^2, where F_e is the set of edges of E^1 that may fail if e so does. For convenience, and in order to follows the notations of Section 4 and 5, we will denote the elements of $\mathcal{F}$ by $F_1, \ldots, F_t$, where $t = |E^2|$ and $G^1 = (V^1, E^1)$ by $G = (V, E)$.

Consider the polytope $\mathcal{P}(G, \mathcal{F})$ given by inequalities (4) and (5). Let x be a solution of $\mathcal{P}(G, \mathcal{F})$. Let $G' = (V', E')$ be a graph obtained by repeated applications of operations $\theta_1, \ldots, \theta_4$. Let $\mathcal{F}' = \{F_1', \ldots, F_t'\}$ where $F_i' = F_i \cap E'$, for $i = 1, \ldots, t$. Let x' be the restriction of x on E'. If x is an extreme point of $\mathcal{P}(G, \mathcal{F})$, then x' is an extreme point of $\mathcal{P}(G', \mathcal{F}')$. In what follows, we show that there is an inequality of type (6) (resp. (11)) (resp. (17)) violated by x in G with respect to $\mathcal{F}$ if and only if there is an inequality of the same type violated by x' in G' with respect to $\mathcal{F}'$.

Given a partition $V_1', \ldots, V_p'$, $p \geq 2$ of V', we let $V_1, \ldots, V_p$ denote the partition of V where V_i, for $i = 1, \ldots, p$, is obtained from V_i' by expanding the nodes of V_i' that arise from the contraction.

Lemma 6.1.

(a) *If x violates an inequality, valid for MSIPND($G, \mathcal{F}$), of type* (6) *(resp.* (11)*) (resp.* (17)*), then x' violates an inequality of type* (6) *(resp.* (11)*) (resp.* (17)*) valid for MSIPND($G', \mathcal{F}'$).*

(b)

(b.1) *If $a'x \geq \alpha'$ is an inequality valid for MSIPND($G', \mathcal{F}'$) of type* (6) *corresponding to $V_1', \ldots, V_p'$ and a set F_i', $i = 1, \ldots, t$, then the inequality $ax \geq \alpha$ where*

$$a(e) = \begin{cases} a'(e) & \text{if} \quad e \in E', \\ 1 & \text{if} \quad e \in \delta_G(V_1, \ldots, V_p) \backslash (F_i \cup E'), \\ 0 & \text{if not}, \end{cases}$$

and $\alpha = \alpha'$ is an inequality of type (6) valid for MSIPND($G, \mathcal{F}$).

(b.2) *If $a'x \geq \alpha'$ is an inequality valid for MSIPND($G', \mathcal{F}'$) of type* (11) *associated with edge sets T_1' and T_2'. Let $T_1 = T_1'$ and $T_2 = T_2' \cup (\delta_G(W) \backslash (T_1' \cup T_2'))$. Let r be as defined in Therorem* 4.6. *Let $ax \geq \alpha$ be the inequality where*

$$a(e) = \begin{cases} r & \text{if} \quad e \in T_2, \\ 1 & \text{if} \quad e \in T_1, \\ 0 & \text{if not}, \end{cases}$$

and $\alpha = \alpha'$. Then $ax \geq \alpha$ is an inequality of type (11) *valid for MSIPND($G, \mathcal{F}$).*

(b.3) *If $a'x \geq \alpha'$ is an inequality valid for MSIPND(G', $\mathcal{F}'$) of type* (17) *associated with a partition $V'_0, \ldots, V'_p$ of V' and sets $F'_{j_1}, \ldots, F'_{j_p}$, valid for MSIPND($G'$, $\mathcal{F}'$). Then the inequality $ax \geq \alpha$ where*

$$a(e) = \begin{cases} a'(e) & \textit{if} \quad e \in E', \\ 1 & \textit{if} \quad e \in ([V_0, V_i] \backslash (F_{j_i} \cup E') \quad \text{for some} \quad i \in \{1, \ldots, p\}, \\ 0 & \textit{if not}, \end{cases}$$

and $\alpha = \alpha'$ is an inequality of type (17) *valid for MSIPND(G, $\mathcal{F}$). Moreover if $a'x \geq \alpha$ is violated by x', then $ax \geq \alpha$ is violated by x.*

Proof:

(a) We will show the statement for the partition inequalities. The proof is similar for the other types of inequalities. For this we may suppose that G' is obtained by one application of either $\theta_1, \theta_2, \theta_3$ or θ_4. Let $V_1, \ldots, V_p$ be a partition of V and $i \in \{1, \ldots, t\}$. And suppose that $x(\delta_{G_i}(V_1, \ldots, V_p)) - p + 1 < 0$.

- If G' is obtained by operation θ_1, it is clear that x' violates the partition inequality in G' corresponding to $\delta_{G'_i}(V_1, \ldots, V_p)$.
- Suppose that G' is obtained by operation θ_2 with respect to an edge f. We claim that $f \notin \delta_{G_i}(V_1, \ldots, V_p)$. Indeed, suppose for instance that $f = uv \in [V_j, V_{j'}] \backslash F_i$, where $j, j' \in \{1, \ldots, p\}$. Let $U = V_j \cup V_{j'}$. As $x(f) = 1$ and $x(e) \geq 0$ for all $e \in [V_j, V_{j'}]$, we have that $x([V_j, V_{j'}]) \geq 0$. Let $\widetilde{V}_1, \ldots, \widetilde{V}_{p-1}$ be the partition of V obtained from $V_1, \ldots, V_p$ by fusionning V_j and $V_{j'}$, in one set, say $\widetilde{V}_1$. Let $\widetilde{V}'_1 = (\widetilde{V}_1 \backslash \{u, v\}) \cup \{w\}$, where w is the node that arises from the contraction of edge f. Notice that $\widetilde{V}'_1, \widetilde{V}_2, \ldots, \widetilde{V}_{p-1}$ is a partition of V'. Moreover we have

$$\begin{aligned} x'(\delta_{G'_i}(\widetilde{V}'_1, \widetilde{V}_2, \ldots, \widetilde{V}_{p-1})) &= x(\delta_{G_i}(\widetilde{V}_1, \ldots, \widetilde{V}_{p-1})) \\ &= x(\delta_{G_i}(V_1, \ldots, V_p)) - x([V_j, V_{j'}] \backslash F_j) \\ &= x(\delta_{G_i}(V_1, \ldots, V_p)) - x(f) - x([V_j, V_{j'}] \backslash (F_j \cup \{f\})) \\ &< p - 2. \end{aligned}$$

  Hence x' violates the partition inequality induced by $\widetilde{V}'_1, \widetilde{V}_2, \ldots, \widetilde{V}_{p-1}$ with respect to F'_i.
- If G' is obtained from G by either θ_3 or θ_4, we can construct in a similar way a partition of G' such that the associated inequality with respect to F'_i is violated by x'.

(b) Observe that all the edges e of $E \backslash E'$ with $x(e) = 1$ have both nodes in the same element of the partition.

□

Lemma (6.1) shows that looking for inequalities of type (6), (11) or (17), which are violated by x, reduces to looking for such inequalities which are violated by x' on G'. Note that this procedure can be applied for any solution of $\mathcal{P}(G, \mathcal{F})$ and, in consequence, may permit to separate fractional solutions which are not necessarly extreme points of $\mathcal{P}(G, \mathcal{F})$. In consequence, for more efficiency, our separation routines will be performed on the reduced graph G'.

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

$$\text{Minimize} \sum_{f \in E} c(f)x(f)$$

$$x(\delta_{G_i}(v)) \geq 1 \qquad \text{for all } v \in V, \qquad \text{for all } i = 1, \ldots, t, \tag{20}$$

$$0 \leq x(f) \leq 1 \qquad \text{for all } f \in E. \tag{21}$$

The optimal solution $y \in \mathbb{R}^E$ of this relaxation of the MSIPND problem is feasible for the problem if y is an integral vector that satisfies all the cut inequalities. Usually, the solution y is not feasible, and thus, in each iteration of the Branch-and-Cut algorithm, it is necessary to generate further inequalities that are valid for the multilayer survivable IP network design problem but violated by the current solution y. For this one has to solve the so-called *separation problem*. This consists, given a class of inequalities, in deciding whether the current solution y satisfies the inequalities, and if not, in finding an inequality that is violated by y. An algorithm which solves this problem is called a *separation algorithm*. For the classes of valid inequalities presented above, the separation is performed in the following order

1. cut constraints
2. cut-cycle constraints
3. star-partition constraints
4. partition constraints

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

Now we describe the separation procedures used in our Branch-and-Cut algorithm. These may be either exact algorithm or heuristics depending on the associated class of inequalities. All our separation algorithms are applied on G' with weights $(\overline{y}'(e), e \in E')$ associated with its edges where $\overline{y}'$ is the restriction on E' of the current LP solution $\overline{y}$.

The separation of the cut constraints for some F_i, $i \in \{1, \ldots, t\}$, can be done in polynomial time using the Gomory-Hu algorithm (Gomory and Hu, 1961). This algorithm produces the so-called Gomory-Hu tree with the property that for all pairs of nodes $s, t \in V'$ the minimum (s,t)-cut in the tree is also a minimum (s,t)-cut in G'. Actually, we use the algorithm developed by Gusfield (1990) which requires $|V'| - 1$ maximum flow computations. The maximum flow computations are handled by the efficient Goldberg and Tarjan algorithm (Goldberg and Tarjan, 1988) that runs in $O(m'n'\log\frac{n'^2}{m'})$ time where m' and n' are the number of edges and nodes of G', respectively. The separation of the cut constraints for each F_i then runs in $O(m'n'^2\log\frac{n'^2}{m'})$ time. The algorithm that permits to separate the cut inequalities for F_i $i = 1, \ldots, t$ is, in consequence, implemented to run in $O(m'n'^2 t\log\frac{n'^2}{m'})$ time.

Algorithm 1

$T_1 \leftarrow \emptyset$; $T_2 \leftarrow \emptyset$; $\bar{\mathcal{F}} \leftarrow \emptyset$;
for $i = 1$ to m **do**
  if $f_i \in F_{j_0}$ for some $j_0 \in \{1, \ldots, t\}$ and $f_i \notin F_j$ for all $F_j \in \bar{\mathcal{F}}$ **then**
    $T_1 \leftarrow T_1 \cup \{f_i\}$;
    $\bar{\mathcal{F}} \leftarrow \bar{\mathcal{F}} \cup \{F_{j_0}\}$;
  else
    $T_2 \leftarrow T_2 \cup \{f_i\}$;
  end if
end for
for all $f_i \in T_2$ **do**
  if $f_i \in F_j$ for all $F_j \in \bar{\mathcal{F}}$ **then**
    $T_2 \leftarrow T_2 \backslash \{f_i\}$;
  end if
end for

Partition inequalities (6) have been shown to be valid for different polyhedra that involve connectivity (Barahona, 1992). A first separation algorithm for these inequalities has been devised by Cunningham (1985) and requires $|E|$ minimum-cut computations. Barahona (1992) reduces this computing time to $|V|$ minimum-cut computations. Both Cunningham and Barahona's algorithm give the most violated inequality. For more efficiency we have implemented Barahona's algorithm which, in consequence, runs in $O(m'n'^3 t \log \frac{n'^2}{m'})$.

To separate the partition inequalities we have developped a faster heuristic. This consists in two steps. In the first step we contract all the edges of E' with value 1. In fact we can remark that if there is a violated partition with some edges with value 1 between some of its elements, then the partition obtained by contracting theses edges is also violated. Then we check whether the trivial partition given by the resulting graph is violated. If not, then we start contracting edges with high value until we get either a graph on p nodes with weight less than $p - 1$ or a graph consisting of two nodes. In the later case, if a violated partition inequality is found, then it is a violated cut inequality.

In our separation we first use this heuristic. If no violated inequality is found, then we try to generate violated partition inequalities using Barahonas's algorithm (Barahona, 1992).

Now we turn our attention to the separation of the cut-cycle inequalities (11). For more efficiency, we have used these constraints only when $q = 1$. From Theorem 4.7 (a) a cut-cycle inequality with $q = 1$, induced by a cut $\delta_G(W)$, which defines a facet for MSIPND(G, $\mathcal{F}$) is of the form

$$x(T_1) + x(T_2) \geq 2, \tag{22}$$

where $T_1 \cap T_2 = \emptyset$, $T_1 \cup T_2 \subset \delta_G(W)$ and T_1, T_2 satisfy Theorem 4.6. To separate constraints of type (22), we have developed a Gomory-Hu tree based heuristic that works as follows.

First we contract all the edges of value 1. Then we compute the Gomory-Hu tree in the resulting graph, say $\widetilde{G}'$. Each cut given by the Gomory-Hu tree, with value less than 2 yields an inequality of type (22) violated by $\overline{y}'$. If $\delta_{\widetilde{G}'}(W) = \{f_1, \ldots, f_m\}$, then sets T_1 and T_2 are determined so that T_1 is maximal, using the following greedy procedure (Algorithm 1).

Since Gomory-Hu algorithm runs with a complexity close to $O(|V'|^4)$, in order to accelerate our separation for the cut-cycle inequalities, we first consider the degree cuts $\delta_{\widetilde{G}'}(v)$, $v \in V'$. The computation of the Gomory-Hu tree on the graph $\widetilde{G}'$ is considered only if no cuts of this type of value less than 2 are found.

We now discuss our separation routine for the star-partition inequalities (17). Our routine consists in determining fractional odd cycles in the supporting graph, satisfying some conditions. In fact, as it can be observed, the graph induced by $\delta(V_1, \ldots, V_p)\backslash F$ where $V_0, \ldots, V_p$ is a violated star-partition with respect to F, may consist of an odd cycle. This is the case, for instance, of the star-partition of the example of Fig.7 (d). This observation, led to the following separation heuristic. We look for odd cycles in G' that are formed by edges whose value is fractional in $\overline{y}'$. Thus, for each detected cycle $(v_1, \ldots, v_p)$ we try to find edge subsets F_{j_i}, $j_i \in \{1, \ldots, t\}$, $i = 1, \ldots, p$ among the edges of $[v_i, V'\backslash\{v_1, \ldots, v_p\}]$ in such way that the star-partition inequality induced by $V'\backslash\{v_1, \ldots, v_p\}, \{v_1\}, \ldots, \{v_p\}$, and F_{j_i}, $i = 1, \ldots, p$ is violated by $\overline{y}'$. This heuristic can be implemented in $O(|V'|^2)$ time.

To store the generated inequalities, we created a pool whose size increases dynamically. All the generated inequalities are put in the pool and are dynamic, i.e. they are removed from the current LP when they are not active. We first separate inequalities from the pool. If all the inequalities in the pool are satisfied by the current LP-solution, we separate the classes of inequalities in the order given above.

7. Computational results

The Branch-and-Cut algorithm described in the previous section has been implemented in C++, using ABACUS (A Branch-And-CUt System) 2.4 alpha (abacus, Elf et al., 2001, Thienel, 1995) to manage the Branch-and-Cut tree and Cplex 8.1 as LP-solver (cplex). It was tested on a Pentium IV 2,4 GHz with 1 Gb RAM, running under Linux. We fixed the maximum CPU time to 5 hours.

Results are presented here for instances coming from real applications and instances obtained from problems of the TSP Library (Reinelt, 1991) by randomly generating the node set and the edge sets F_e. For all the instances, the graph G^1, representing the IP network, is considered complete.

The edge costs for the random instances are equal to rounding Euclidian distances. These instances were generated with 10 to 45 nodes and $|\mathcal{F}| = 10, 15, 20, 30$. Five instances of each size and each $|\mathcal{F}|$ were tested. We will consider the average results obtained for these instances.

The real instances are extracted from operational networks and have been provided by the french telecommunications operator France Télécom. These instances have 18 to 60 nodes and $\mathcal{F}$ with 31 to 102 edge sets. Actually France Télécom has provided the optical network and the routing between every pair of nodes in this network. With an edge f of the IP network, we associate the routing path of the optical network between the switches corresponding to the IP routers extremities of f. Using these paths we have computed $\mathcal{F} = \{F_e \subseteq E^1,\ e \in E^2\}$ where F_e is the set of edges f of E^1 such that e belongs to the path associated with f. For these instances we have used different cost functions.

In the various tables, the entries are:

$\lvert V^1 \rvert$	:	the number of nodes of G^1,
$\lvert\mathcal{F}\rvert$	:	the number of sets F_e,
NC	:	the number of generated cut inequalities,
NCC	:	the number of generated cut-cycle inequalities,
NSP	:	the number of generated star-partition inequalities,
NP	:	the number of generated partition inequalities,
NT	:	the number of generated nodes in the Branch-and-Cut tree,
o/p	:	the number of problems solved to optimality over the number of instances tested (only for random instances),
Copt	:	the value of the optimal solution (only for real instances),
Gap	:	the relative error between the best upper bound (the optimal value if the problem has been solved to optimality) and the lower bound achieved by the cutting plane phase (before branching),
TT	:	the total CPU time in h:mm:ss.

Our first series of experiments concern the random instances. Table 1 reports the average results obtained for these instances. We remark that for 25 nodes or less, all problems could be solved to optimality. Moreover for $|\mathcal{F}| = 10$ all instances have been solved to optimality within the time limit. For the instances of 30 nodes (resp. 35 nodes) with $|\mathcal{F}| \leq 20$ (resp. $|\mathcal{F}| \leq 15$) we could also get the optimal solution for all the tested instances. However for $|\mathcal{F}| = 30$ (resp. $|\mathcal{F}| = 20$), only four over the five tested instances have been solved.

The problems with 35 nodes and more and $|\mathcal{F}| \geq 20$ seems to be harder to solve. In fact we can remark that for 40 and 45 nodes and $|\mathcal{F}| \geq 20$, no more that two instances have been solved in the time limit. For $|\mathcal{F}| = 30$, none of the instances have been solved to optimality. We may also observe that, in general, the problem gets harder when $|\mathcal{F}|$ increases, which explains the rising of the CPU time with respect to $|\mathcal{F}|$. This is quite natural since for a given $\mathcal{F}$ we have to solve $|\mathcal{F}|$ connected subgraph problems.

We can also note that, for most of the instances, a significant number of cut, cut-cycle and partition inequalities have been generated. This implies that these inequalities are useful for the random problems. However, the star-partition inequalities do not seem to play an important role for this type of instances. This can be explained by the fact that, for these instances, the edges do not necessarily belong to some F_e's, and hence, it could be hard to find star-partitions satisfying the conditions of Theorem 4.8.

Finally, we notice that for many problems, our constraints have not been sufficient to solve the problem in the cutting plane phase. We can observe, however, that the gap is relatively small when $|\mathcal{F}| \leq 20$. (The gap does not appear in the table for the problems where no upper bound could be obtained for one of the five tested instances). Tables 2–4 present results for the real instances obtained for different cost functions. Usually the cost associated with a link in the client network is related to the corresponding routing path in the optical network, and then depends on the cost of this path. Actually, the cost $c(f)$ of link f in the IP network is given by

$$c(f) = c + \omega(f),$$

where c is a fixed cost representing the equipments of the extremity ports on the routers of f in the IP layer, and $\omega(f)$ is a cost depending on the lenght of the path P_f corresponding to f in the optical network.

Table 1 Results for random instances

$\|V^1\|$	$\|\mathcal{F}\|$	NC	NCC	NSP	NP	NT	o/p	Gap	TT
10	10	59.6	1.0	0.0	28.0	9.8	5/5	1.80	0:00:00.88
10	15	40.8	1.2	0.2	38.8	5.4	5/5	0.92	0:00:01.03
10	20	50.6	1.6	0.2	46.8	2.6	5/5	0.32	0:00:01.35
10	30	66.0	2.4	0.0	62.6	7.0	5/5	1.39	0:00:02.61
15	10	185.4	2.4	0.0	71.8	15.0	5/5	1.11	0:00:05.06
15	15	176.6	3.0	0.2	86.2	11.8	5/5	2.13	0:00:08.63
15	20	158.0	3.6	0.2	90.0	10.6	5/5	1.25	0:00:10.33
15	30	124.2	9.2	0.0	198.0	43.8	5/5	3.43	0:00:51.39
20	10	194.8	2.0	0.2	216.6	45.8	5/5	1.57	0:00:41.08
20	15	271.8	5.8	0.0	300.8	99.0	5/5	2.82	0:02:03.77
20	20	230.2	10.0	0.2	885.6	294.2	5/5	3.69	0:08:58.00
20	30	264.2	16.2	0.2	1897.8	575.8	5/5	5.16	0:26:17.80
25	10	315.6	3.2	0.2	328.6	71.8	5/5	1.65	0:02:29.52
25	15	281.4	10.0	0.6	924.0	236.6	5/5	2.56	0:12:02.02
25	20	295.0	10.2	0.2	879.6	140.2	5/5	2.46	0:11:34.43
25	30	320.6	20.6	0.0	2338.4	792.2	5/5	4.86	0:59:55.25
30	10	419.4	2.4	0.0	364.8	27.0	5/5	0.58	0:02:22.14
30	15	485.4	5.4	0.2	750.2	1406.3	5/5	1.55	0:10:25.21
30	20	485.6	7.0	0.2	941.4	143.4	5/5	2.02	0:17:36.16
30	30	352.6	20.0	0.6	3259.2	654.07	4/5	–	1:48:30.62
35	10	473	4.2	0.2	787.4	79.8	5/5	1.22	0:13:48.24
35	15	492	10.0	1.0	2958.0	527.8	5/5	2.35	1:38:06.83
35	20	461.6	649.4	1.0	3042.8	491.0	4/5	–	2:55:06.68
35	30	398.8	396.8	0.0	3676.0	502.2	0/5	–	5:00:00.00
40	10	605.8	2.6	0.4	776.0	113.8	5/5	1.05	0:20:37.71
40	15	574.8	11.6	1.4	2102.4	382.6	4/5	–	1:37:53.85
40	20	511.4	22.4	2.4	3323.2	458.6	2/5	–	3:11:11.58
40	30	522.0	518.2	0.8	3070.4	257.4	0/5	–	5:00:00.00
45	10	562.6	5.8	0.4	1270.6	189.8	5/5	1.28	0:57:51.81
45	15	585.0	30.6	1.8	2830.8	333.8	2/5	–	3:31:00.00
45	20	498.4	132.2	0.4	3740.4	236.2	0/5	–	5:00:00.00
45	30	548.2	137.4	0.8	2504.2	129.8	0/5	–	5:00:00.00

The installation of an optical segment usually yields a fixed cost on each extremity of this segment. Hence a first estimation of the optical cost $\omega(f)$ is the sum of the fixed costs of the optical segments on P_f. As these fixed costs can be considered the same in the optical network, a good approach would be to consider a cost $\omega(f)$ proportitional to the number of the optical segments on P_f. So, a first natural function $\omega(f)$ consists of the number of links (hops) in the optical path between the switching nodes corresponding to the extremities of f. Here we assume that there is a fixed cost associated with each optical link. This cost is considered once the corresponding link is used. Then the cost $c(f)$ is given in this case by $c_1(f) = c + |P_f|$

Table 2 Results for real instances with the cost function $c_1(.)$

$\|V^1\|$	$\|\mathcal{F}\|$	NC	NCC	NSP	NP	NT	Copt	Gap	TT
18	31	70	5	5	31	1	111	0.00	0:00:11.72
25	39	32	3	1	39	1	150	0.00	0:00:24.83
25	40	44	5	17	68	11	154	0.00	0:01:12.98
28	45	311	15	5	762	1	168	0.00	0:05:24.17
28	46	176	10	13	189	3	171	0.00	0:01:45.34
30	55	176	14	18	220	1	185	0.00	0:02:43.25
31	46	133	0	0	46	1	188	0.00	0:00:29.93
32	56	180	17	39	1182	17	196	0.00	0:19:41.83
33	54	74	6	1	54	1	198	0.00	0:02:28.36
34	62	53	6	0	62	1	206	0.00	0:01:33.73
35	63	614	17	110	2678	705	*223*	3.14	5:00:00.00
36	68	581	15	9	68	1	218	0.00	0:07:19.62
38	65	303	19	13	249	11	229	0.00	0:15:57.94
40	76	432	24	40	1263	23	245	0.00	1:03:55.41
41	73	453	21	45	977	19	252	0.00	0:38:51.53
42	70	546	25	8	937	9	254	0.00	0:35:03.26
43	74	635	20	11	252	11	260	0.00	0:31:20.90
45	78	468	24	7	319	9	272	0.00	0:40:24.27
45	86	253	16	13	186	9	274	0.00	0:33:11.47
46	85	652	25	38	432	37	281	0.00	1:11:58.58
47	84	878	17	2	84	1	282	0.00	0:25:29.03
48	91	851	29	43	852	43	291	0.00	1:52:48.22
49	88	342	26	24	128	7	295	0.00	1:07:56.12
50	83	725	33	34	2539	67	304	0.33	2:53:14.94
50	95	419	20	27	191	7	304	0.33	0:56:53.31
55	95	768	34	14	2689	5	334	0.00	2:58:26.05
60	102	588	37	23	2335	17	–	–	5:00:00.00

Table 2 gives results obtained when the cost of each edge f of the IP network is given by $c_1(f)$. We remark that all the instances have been solved to optimality except those with 35 and 60 nodes. For most of these instances, the gap is 0. However the algorithm needed to branch for obtaining a feasible optimal solution.

All the instances with 38 nodes and less, except that of 35 nodes, have been solved to optimality in less than 20 minutes. The rest of the instances, except that with 60 nodes, could be solved in less than 3 hours.

For the instance on 35 nodes, a feasible solution of value 223 (given in italic) has been obtained, yielding a relatively small gap of 3.14%. For the 60 nodes instance, we could not obtain an upper bound (feasible solution) within the time limit. Network operators often consider that the number of optical segments on a path is strongly related to the path lenght in km's. Therefore, they propose path cost models linearly dependent of the kilometric path length. In this case, $c(f)$ can then be defined as

$$c_2(f) = c + \sum_{e \in P_f} l(e),$$

where $l(e)$ is the lenght of e. For our purpose, we consider $l(e)$ as the real distance (in km's) between the extremities of e.

Table 3 Results for real instances with the cost function $c_2(.)$

$\lvert V^1\rvert$	$\lvert\mathcal{F}\rvert$	NC	NCC	NSP	NP	NT	Copt	Gap	TT
18	31	84	5	4	28	101	14999.70	3.13	0:00:51.46
25	39	82	5	0	0	1	8968.24	0.00	0:00:14.73
25	40	127	6	18	0	19	17538.70	0.03	0:01:10.05
28	45	113	8	3	71	3	10764.60	0.05	0:01:02.66
28	46	425	12	37	218	163	18035.50	2.57	0:12:57.25
30	55	25	5	14	0	1	19554.30	0.00	0:00:45.30
31	46	103	3	6	0	1	5775.00	0.00	0:00:30.62
32	56	236	15	60	977	405	19406.60	2.83	1:17:58.67
33	54	200	9	2	85	3	12472.80	0.04	0:01:36.00
34	62	386	11	18	151	9	18466.90	0.71	0:09:00.50
35	63	62	9	34	0	25	20574.30	0.10	0:07:47.10
36	68	128	5	6	0	1	18061.30	0.00	0:01:30.68
38	65	226	16	5	258	13	14778.90	0.06	0:08:52.99
40	76	377	11	31	294	45	21048.10	1.38	0:41:04.27
41	73	283	8	10	0	1	21137.80	0.00	0:04:16.04
42	70	335	22	34	368	21	6164.22	0.51	0:29:58.31
43	74	324	16	7	327	1	16536.30	0.00	0:13:46.26
45	78	412	16	10	245	1	16768.20	0.00	0:17:00.81
45	86	101	1	0	0	1	1152.79	0.00	0:01:15.16
46	85	484	10	26	4	15	22028.80	0.53	0:20:34.91
47	84	656	21	4	294	1	16762.20	0.00	0:21:34.26
48	91	510	9	22	4	15	20811.20	0.56	0:28:01.64
49	88	667	22	4	88	1	17115.50	0.00	0:20:18.75
50	83	573	14	29	620	103	6699.38	0.40	1:58:09.95
50	95	485	8	6	333	9	21847.50	0.05	0:21:47.15
55	95	227	18	24	95	23	6975.36	0.06	1:24:12.13
60	102	785	31	22	846	87	–	–	5:00:00.00

Table 3 reports the results for the real instances with the cost function $c_2(.)$. We note that all the instances, except the instance with 60 nodes, have been solved to optimality and in less than two hours. Some of the instances with less than 40 nodes have even been solved in less than 10 minutes. We can also see that the gap does not exceed 1.5% for most of the instances.

The last cost function used for the real instances is defined as follows. Let $f = uv$ be an edge of G^1. Let u' and v' be the nodes of G^2 corresponding to u and v, respectively. Let

$$c_3(f) = \begin{cases} \frac{c_2(f)}{|F_e|} & \text{if } e = u'v' \in E^2, \\ c_2(f) & \text{if not.} \end{cases}$$

The cost function $c_3(.)$ depends on the number ($|F_e|$) of the routing paths that use $e = u'v'$ in the optical network, if $e \in E^2$. The motivation to consider this is that when $|F_e|$ is high, the link f could be essential in the IP network. For this, $c_3(f)$ gets small, and hence one may hope that f appears in the optimal solution. The results obtained using this cost function are presented in Table 4. Here it seems that the problem is much easier to solve. In fact, all the instances have been solved to optimality in less than 10 minutes and the resolution of all the instances except that with 28 nodes and $|\mathcal{F}| = 46$ have been achieved in the cutting plane phase. We may also remark that no partition and no star-partition inequalities have

Table 4 Results for real instances with the cost function $c_3(.)$

$\|V^1\|$	$\|\mathcal{F}\|$	NC	NCC	NSP	NP	NT	Copt	Gap	TT
18	31	33	0	0	0	1	1573.180	0.00	0:00:00.60
25	39	5	0	0	0	1	584.670	0.00	0:00:02.73
25	40	0	0	0	0	1	1518.800	0.00	0:00:01.15
28	45	50	0	0	0	1	791.918	0.00	0:00:03.59
28	46	148	2	0	1	3	1111.630	0.27	0:00:26.68
30	55	11	0	0	0	1	1701.370	0.00	0:00:07.54
31	46	6	0	0	0	1	526.308	0.00	0:00:08.19
32	56	12	0	0	0	1	1027.510	0.00	0:00:09.58
33	54	62	1	0	0	1	1293.830	0.00	0:00:14.49
34	62	85	1	0	0	1	1055.990	0.00	0:00:32.75
35	63	8	0	0	0	1	1590.780	0.00	0:00:22.67
36	68	6	0	0	0	1	1053.930	0.00	0:00:13.64
38	65	69	3	0	0	1	1300.670	0.00	0:00:45.73
40	76	0	0	0	0	1	1077.520	0.00	0:00:08.29
41	73	23	1	0	0	1	1455.400	0.00	0:01:16.68
42	70	13	3	0	0	1	184.324	0.00	0:01:00.03
43	74	130	5	0	0	1	1600.370	0.00	0:03:18.41
45	78	185	7	0	0	1	1522.030	0.00	0:03:24.01
45	86	101	1	0	0	1	1152.790	0.00	0:01:15.16
46	85	12	0	0	0	1	1029.461	0.00	0:00:45.42
47	84	546	10	0	0	1	1545.770	0.00	0:05:08.76
48	91	15	0	0	0	1	701.019	0.00	0:01:21.16
49	88	326	4	0	0	1	971.864	0.00	0:05:24.76
50	83	100	2	0	0	1	186.281	0.00	0:03:05.83
50	95	0	0	0	0	1	716.747	0.00	0:00:24.12
55	95	0	1	0	0	1	185.075	0.00	0:01:39.83
60	102	140	6	0	0	1	166.458	0.00	0:10:08.39

been generated for all of the instances except that of 28 nodes. For this last instance, we got a very small gap of 0.27%.

From these three tables that the difficulty to solve the real instances heavily depends on the cost function associated with the IP network. It seems that, as far as this function becomes uniform, the problem gets degenerate and hence harder to solve. This is what may explain the difference between the results of the tables.

For the cost functions $c_1(.)$ and $c_2(.)$, a significant number of cut-cycle and star-partition inequalities have been generated. In order to evaluate the impact of these inequalities as well as the reduction operations $\theta_1, \ldots, \theta_4$ on the performance of the algorithm, we report in Table 5 results obtained for real instances without using neither these inequalities nore the reduction operations.

For the cost function $c_1(.)$, five instances in Table 5 could not be solved in the time limit of 5 hours. By using the reduction operations and adding the cut-cycle and star-partition inequalities, four over the five instances have been solved in less than 2 hours. And for the remaining one, as indicated in Table 2, we could get a feasible solution. Also for instance with 30 nodes, 33 nodes have been generated in the Branch-and-Bound tree, whereas the resolution of this instance, in Table 2, has been achieved in the cutting plane phase. For the second set of instances in Table 5, which use the cost function $c_2(.)$, the problem tests, that have been solved to optimality using the cut-cycle and the star-partition inequalities, are also solved to optimal-

Table 5 Results for real instances without cut-cycle and star-partition inequalities

Cost functions	$\|V^1\|$	$\|\mathcal{F}\|$	NC	NP	NT	Copt	Gap	TT
$c_1(.)$	30	55	229	1133	33	185	0.54	0:09:42.13
$c_1(.)$	35	63	213	6404	127	–	–	5:00:00.00
$c_1(.)$	41	73	231	397	17	–	–	5:00:00.00
$c_1(.)$	46	85	305	6942	43	–	–	5:00:00.00
$c_1(.)$	48	91	307	5248	65	–	–	5:00:00.00
$c_1(.)$	50	95	297	3305	143	–	–	5:00:00.00
$c_2(.)$	30	55	128	204	37	19554.3	0.05	0:03:36.32
$c_2(.)$	31	46	148	130	15	5775.0	0.09	0:01:58.04
$c_2(.)$	35	63	196	706	295	20574.3	0.41	0:47:31.20
$c_2(.)$	41	73	267	765	39	21137.8	0.18	0:25:31.08
$c_2(.)$	48	91	229	330	47	20811.2	0.71	0:47:23.91
$c_2(.)$	50	95	285	1870	71	21847.5	1.34	2:07:04.73
$c_3(.)$	28	46	69	23	7	1111.6	1.15	0:00:37.50
$c_3(.)$	41	73	24	0	5	1455.4	0.02	0:02:13.68

ity without using these inequalities, but the computing time more than doubled. For instance, the problem with 50 nodes which is solved in less than 22 minutes, with the cut-cycle and the star-partition inequalities, needed more than 2 hours without these inequalities. Also we can note that the size of the Branch-and-Bound tree has been significantly increased for all the instances. Observe that the instances with 30, 31 and 41 nodes are solved in Table 3 without any branching.

The two last problems of Table 5 correspond to the cost function $c_3(.)$. We can also remark that the cut-cycle inequalities have been useful for solving these problems.

Finally, let us note that many of the partition, cut-cycle and star-partition inequalities, that cut off fractional solutions in the experiments, were facet defining in G' and also in G because of the application of the reduction operations.

8. Concluding remarks

In this paper we have considered a multilayer survivable network design problem that may have applications to the design of reliable IP-over-optical network. We have proposed an integer programming formulation for the problem and studied a cutting plane approach for solving it. We have identified some valid inequalities, and describe necessary conditions and sufficent conditions for these inequalities to define facets. We have also discussed separation routines and introduced some reduction operations defined with respect to a fractional solution of the linear relaxation. Using these results we have described a Branch-and-Cut algorithm for the problem. Our computational results have shown that the reduction operations play a central role for accelerating the separation process, and the cut-cycle and star-partition inequalities are very effective for the problem.

In addition to the survivability aspect, one can consider the capacity dimensioning of the network. These issues have been treated simultaneously in the literature from different point of views, in both the monolayer case (Magnanti, Mirchandani, and Vachani, 1993, 1995; Stoer and Dahl, 1994; Wessäly, 2000) and the multilayer one (Gouveia et al., 2003; Ricciato et al., 2002). An interesting question would be to extend the study developed in this paper to the more general capacitated multilayer model.

Acknowledgments We thank the anonymous referees for their valuable comments.

References

Abacus. http://www.oreas.de.

Barahona, F. (1992). "Separating from the Dominant of the Spanning Tree Polytope." *Operations Research Letters*, 12, 201–203.

Barahona, F. and A.R. Mahjoub. (1995). "On Two-Connected Subgraph Polytopes." *Discrete Mathematics*, 147, 19–34.

Bradner, S. IETF. (1995). "The Recommendation for the IP Next Generation Protocol." (RFC1752).

Cplex. http://www.ilog.com/products/cplex/.

Cunningham, W.H. (1985). "Optimal Attack and Reinforcement of a Network." *Journal of Association for Computing Machinery*, 32, 549–561.

Elf, M., C. Gutwenger, M. Jünger, and G. Rinaldi. (2001). "Branch-and-Cut Algorithms for Combinatorial Optimization and Their Implementation in ABACUS." In M. Jünger, and D. Naddef (eds.), *Computational Combinatorial Optimization*, LNCS 2241, pp. 157–222.

Goldberg, A.V. and R.E. Tarjan. (1988). "A New Approach to the Maximum-flow Problem." *Journal of the Association for Computing Machinery*, 35, 921–940.

Gomory, R.E. and T.C. Hu. (1961). "Multi-Terminal Network Flows." *SIAM Journal on Applied Mathematics*, 9, 551–570.

Gouveia, L., P. Patrício, A. de Sousa, and R. Valadas. (2003). "MPLS Over WDM Network Design with Packet Level QoS Constraints Based on ILP Models." In *Proceedings IEEE INFOCOM 2003*, San Francisco.

Grötschel, M., C.L. Monma, and M. Stoer. (1995). "Design of Survivable Networks." In M.O. Ball et al. (eds.), *Handbooks in OR & MS 7*, pp. 617–671.

Gusfield, D. (1990). "Very Simple Methods for All Pairs Network Flow Analysis." *SIAM Journal on Computing*, 19, 143–155.

Jensen, T. (2001). "Internet Protocol and Transport Protocols." *Telektronikk*, 97, 20–38.

Kerivin, H. (2000). "Réseaux fiables et polyèdres." PhD Thesis, University Blaise Pascal, Clermont-Ferrand, France.

Kerivin, H., A.R. Mahjoub, and C. Nocq. (2004). "(1, 2)-Survivable Networks: Facets and Branch-and-Cut." In M. Grötschel (ed.), *The Sharpest Cut, MPS/SIAM in Optimization*, pp. 121–152.

Magnanti, T.L., P. Mirchandani, and R. Vachani, (1993). "The Convex Hull of Two Core Capacitated Network Design Problems." *Mathematical Programming*, 60, 233–250.

Magnanti, T.L., P. Mirchandani, and R. Vachani. (1995). "Modeling and Solving the Two-Facility Capacitated Network Loading Problem." *Operations Research*, 43, 142–157.

Mahjoub, A.R. (1994). "Two-Edge Connected Spanning Subgraphs and Polyhedra." *Mathematical Programming*, 64, 199–208.

Postel, J. IETF. (1981). "Internet Protocol. DARPA Internet Program Protocol Specification." (RFC791).

Postel, J. IETF. (1981)."Internet Control Message Protocol. DARPA Internet Program Protocol Specification." (RFC792).

Postel, J. IETF. (1981). "Transmission Control Protocol. DARPA Internet Program Protocol Specification." (RFC793).

Rajagopalan, B., D. Pendarakis, D. Saha, R.S. Ramamoorthy, and K. Bala. (2000). "IP Over Optical Networks: Architectural Aspects." *IEEE Communications Magazine*, 94–102.

Reinelt, G. (1991). "TSPLIB—A Traveling Salesman Problem Library." *ORSA Journal on Computing*, 3, 376–384.

Ricciato, F., S. Salsano, A. Belmonte, and M. Listanti. (2002). "Off-line Configuration of a MPLS over WDM Network under Time-Varying Offered Traffic." In *Proceedings IEEE INFOCOM 2002*, New York, USA.

Schrijver, A. (2003). "Combinatorial Optimization: Polyhedra and Efficiency." *Algorithms and Combinatorics*, 24, Springer-Verlag, Berlin.

Stoer, M. (1992). "Design of Survivable Networks." *Lectures Notes in Mathematics 1531*, Springer-Verlag.

Stoer, M. and G. Dahl. (1994). "A Polyhedral Approach to Multicommodity Survivable Network Design." *Numerische Mathematik*, 68, 149–167.

Thienel, S. (1995). "ABACUS—A Branch-And-CUt System." PhD thesis, Universität zu Köln.

Wessäly, R. (2000). "DImensioning Survivable Capacitated NETworks(DISCNET)." PhD Thesis, Technische Universität Berlin.

Zouganeli, E. (2001). "Optical Network Functionality: From "Dumb Fat Pipes" to Bright Networking." *Telektronikk*, 97, 346–354.