The Multilayer Capacitated Survivable IP Network Design Problem : valid inequalities and Branch-and-Cut

Sylvie Borne, Eric Gourdin, Olivier Klopfenstein, Ridha Mahjoub

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
CNRS FRE 3234

CAHIER DU LAMSADE

300

Septembre 2010

The Multilayer Capacitated Survivable IP Network Design Problem : Valid inequalities and Branch-and-Cut

S. BORNE, E. GOURDIN, O. KLOPFENSTEIN, A. R. MAHJOUB

The Multilayer Capacitated Survivable IP Network Design Problem : valid inequalities and Branch-and-Cut

S. Borne[1*], E. Gourdin[2], Olivier Klopfenstein[2] and A. R. Mahjoub[1†]

[1] *Laboratoire LIMOS, CNRS, Université Blaise Pascal - Clermont Ferrand II*
Complexe scientifique des Cézeaux, 63177 Aubière Cedex, France
sylvie.borne@lipn.univ-paris13.fr, mahjoub@lamsade.dauphine.fr
[2] *Orange Labs Research & Development*
38-40 rue du Général-Leclerc, 92794 Issy-les-Moulineaux Cedex 9, France
(eric.gourdin,olivier.klopfenstein)@orange-ftgroup.com

Abstract : Telecommunication networks can be seen as the stacking of several layers like, for instance, IP-over-Optical networks. This infrastructure has to be sufficiently survivable to restore the traffic in the event of a failure. Moreover, it should have adequate capacities so that the demands can be routed between the origin-destinations. In this paper we consider the Multilayer Capacitated Survivable IP Network Design problem. We study two variants of this problem with simple and multiple capacities. We give two multicommodity flow formulations for each variant of this problem and describe some valid inequalities. In particular, we characterize valid inequalities obtained using Chvatal-Gomory procedure from the well known Cutset inequalities. We show that some of these inequalities are facet defining. We discuss separation routines for all the valid inequalities. Using these results, we develop a Branch-and-Cut algorithm and a Branch-and-Cut-and-Price algorithm for each variant and present extensive computational results.

Keywords : IP-over-optical network, survivability, capacities, Branch-and-Cut-and-Price algorithms.

1 Introduction

In the past years, telecommunication 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 information. Hence, in the event of a catastrophic failure, a big amount of traffic may be lost. In consequence telecommunication 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 telecommunication 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 telecommunication 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,

* Present address : Laboratoire LIPN, UMR CNRS 7030, Institut Galilée, Université Paris 13
99 Avenue J-B. Clément, 93430 Villetaneuse, France
† Present address : Laboratoire LAMSADE, UMR CNRS 7024, Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France

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 independently 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 to install on the different layers of a network, usually called *dimensioning*, 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 telecommunication (like GMPLS) [44] 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 [43], Voge studies different problems about the multilayer telecommunication networks based on MPLS and GMPLS.

In this paper we introduce a multilayer capacitated survivable network design problem that may be of practical interest for the design and the dimensioning of IP-over-optical networks. These networks, based on the GMPLS technology, consist of two layers, the IP (service, client) layer and the optical (transport) layer. We give mixed integer programming formulations for this problem and discuss Branch-and-Cut and Branch-and-Cut-and-Price algorithms.

Survivability and dimensioning have already been studied in the literature for multilayer networks. In particular, heuristic approaches have been proposed. In [18, 19], 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 [39], 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. There are also some recent works carried out on two-layered network design with or without dimensioning. In [34], Orlowsky and Wessäly describe a general integer linear programming model for the design of multi-layer telecommunication network de sign problem which integrates hardware, capacity, routing and grooming decisions. They give also a sketch of an algorithmic approach. Orlowski et al. [33] develop three primal heuristics to be called in a Branch-and-Cut algorithm to solve the problem with two layers. Knippel and Lardeux [27] study heuristic and exact algorithms based on metric inequalities for a multilayer design problem (see also [28]).

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 [3, 20, 25, 26, 32, 41].

Most of research on the design of networks concentrates on uncapacitated networks where each link can support all the traffic at once. However, for many telecommunication networks, capacities play a fundamental role. Nevertheless several problems with capacity loading have been studied. In

[13] Dahl and Stoer present a cutting plane approach for solving the MULTIcommodity SUrvivable Network design problem (MULTISUN problem). This consists, given point-to-point traffic demands in a network, in finding minimum cost capacities that permit the routing of the given demands. The possible capacity choices on each edge give rise to a discrete cost function. Another problem called the Network Loading Problem (NLP) plays a central role in the design of telecommunication networks. It is a special case of the MULTISUN problem. For the NLP, a single type of capacitated facility is considered, and each link can be assigned one or several facilities in such a way that the network can carry given point-to-point demands at minimum cost. Barahona [4] studies this problem in the both nonbifurcated and bifurcated cases that is when the flow of each commodity is carried by a single path or when it could use several paths. He proposes a separation algorithm for the so-called cut inequalities, which seem to play a central role for solving the problem. In [30] Magnanti et al. study the polyhedral structure of two core subproblems of the NLP. And in [31] they discuss a further problem, called the Two-Facility capacitated network Loading Problem (TFLP), in which considers two types of capacities.

The paper is organised as follows. In the following section we discuss the IP-over-optical networks and the interaction between the different layers. We present a multilayer survivable network design problem with capacity constraint, called the multilayer capacitated survivable IP network design problem. We describe two versions of this problem : with and without multiple edges. We give mixed integer programming formulations for this problem. Section 3 presents the column generation algorithm to solve a linear relaxation of the problem. In Section 4, we study the associated polytopes. We identify a few classes of valid inequalities and describe conditions for theses inequalities to be facet defining. In Section 5, we describe the Branch-and-Cut algorithms and Branch-and-Cut-and-Price algorithms for the problem. Our computational results are presented and discussed in Section 6. In Section 7, we give some concluding remarks.

2 The Multilayer Capacitated Survivable IP Network Design Problem

2.1 Multilayer telecommunication networks

Telecommunication 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 [9, 24, 35, 36, 37].

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 between two routers in the IP network 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 seven switches $S_1, \ldots, S_7$. Only the optical switches $S_1, \ldots, S_4$ communicate with one router throught the UNI.

The introduction of this new infrastructure of telecommunication networks gives rise to survivability issues. 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 $S_1 - S_2$, and the link $R_1 - R_3$ corresponds to the path $S_1 - S_2 - S_6 - S_3$. Here, the network is not survivable to single link failures. For instance, if the optical link $S_1 - S_2$ fails, then the links in the IP network $R_1 - R_2$ and

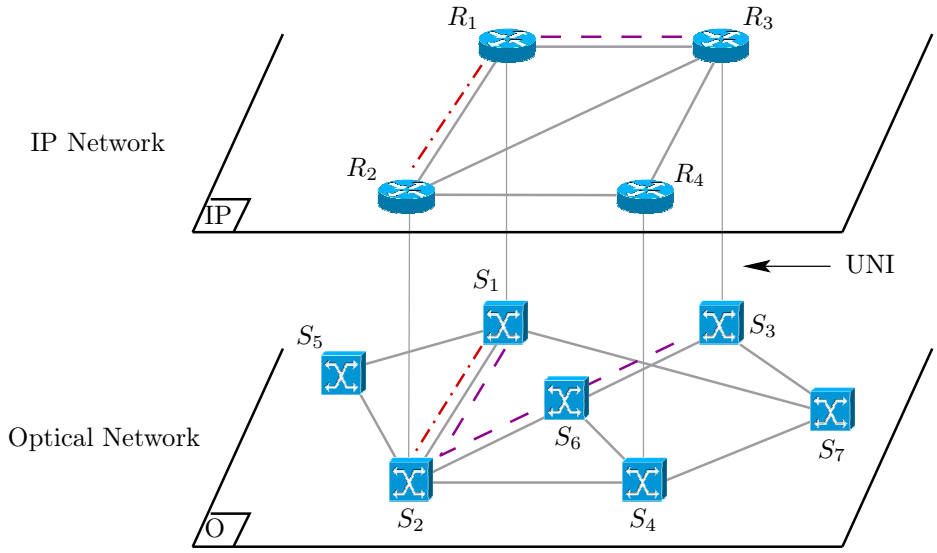


FIG. 1 – An IP-over-optical network

$R_1 - R_3$ are cut, and therefore the router R_1 is no more connected to the rest of the routers. As a 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.

In addition to the survivability aspect, we may need to install capacities on the IP network in order to route commodities between some routers. In this paper we shall be concerned by this problem which considers simultaneously both the survivability and the dimensioning of the IP network when the transport network is fixed.

2.2 The problem

The first major survivability requirement used in telecommunication networks is the 2-connectivity. That is there must exist at least two edge-disjoint paths between every pair of nodes in the network. This assumption, that only one edge may fail at a time, is based on the naive idea that the links in the network are independent 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. As a result, several edges may fail at the same time in the IP layer (such a group of links is usually refered to as a Shared Risk Link Group, or SRLG).

The multilayer survivable IP network design problem (MSIPND problem) introduced by Borne et al. [7] consists in finding 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 our problem, we can install capacities of 2.5 Gbits or 10 Gbits on any link of the IP network. Usually, the installed capacities have to be symmetric. Hence, we consider that each time a certain capacity is installed from a router R_1 to a router R_2, one has to install the same capacity from R_2 to R_1. Figure 2 shows the eligible and the forbidded situations.

Consequently, a more realistic model which has to be investigated, would consist in setting up

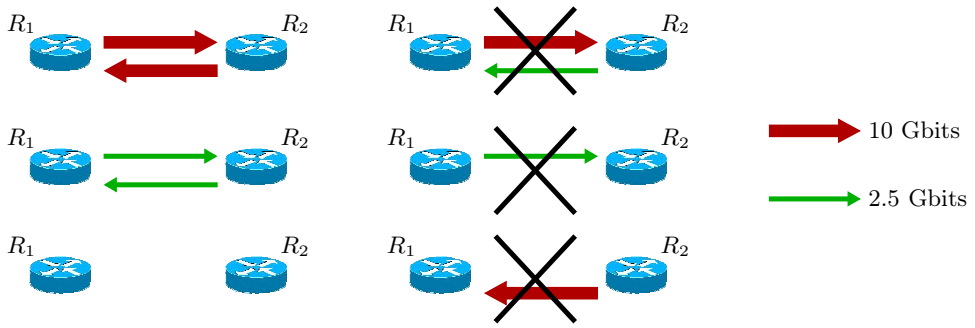


FIG. 2 – Symmetric capacities

capacities with a minimum cost to the client network that allows a multicommodity flow which respects the capacities for any simple edge failure in the transport network.

In this paper we consider this problem. More precisely, we consider the overlay model where the IP and the optical networks are separated. We suppose that the topology and the routing of the optical network are fixed and satisfy some survivability requirements. We also 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 type of capacity on each edge of the IP network, the *Multilayer Capacitated Survivable IP Network Design problem* (MCSIPND problem) is to find the minimum cost set of links to be installed in the IP network and facilities to be loaded on theses links so that if a failure occurs on an optical link, the IP subnetwork obtained by removing the corresponding edges allows a multicommodity f low which satisfies the capacities.

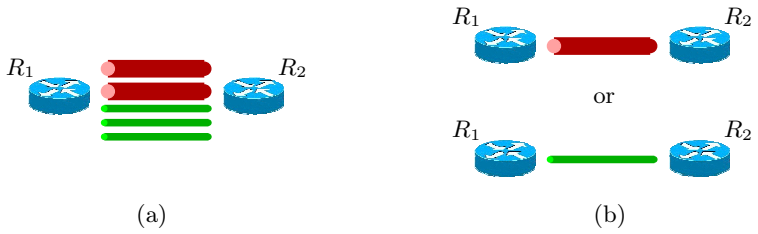


FIG. 3 – Two variants of the problem

We consider two variants of the problem : the multiple MCSIPND (denoted by MCSIPND$_m$) which allows multiple links, and the simple MCSIPND (denoted by MCSIPND$_s$) where only one link of one type of capacity can be loaded between two routers. Figure 3 illustrates the two possibilities of installing capacities. The orientation of the links between the routers is omitted because the capacities are symmetric. A capacity installed in the two directions between two routers is represented by a cylinder (a small one for capacity 2.5 Gbits and a big one for 10 Gbits). Figure 3(a) corresponds to the multiple case and figure 3(b) to the simple one.

In what follows we give mixed integer programming formulations for the MCSIPND problem. To this end, we first introduce some definitions and notations.

2.3 Definitions and notation

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. We denote also by $D = (V, A)$ the bidirected graph associated with G such that each edge $e = uv \in E$ is replaced by two arcs (u, v) and (v, u), respectively from u to v and from v to u, in the *arc set* A of D. For an

edge subset $F \subseteq E$ we denote by $\vec{F} \subseteq A$ the associated arc subset. For $F \subseteq E$ we let $G\backslash F$ denote the subgraph of G obtained by removing the edges of F and $D\backslash\vec{F}$ the associated subgraph obtain from D by removing the arcs of $\vec{F}$. Throughout the paper we will consider simple graphs.

Let $G = (V, E)$ be an undirected graph. 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. For $W \subseteq V$, we denote by $G(W)$ the subgraph of G induced by W. If $W \subset V$, $\overline{W}$ denotes $V\backslash W$. 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.

Given a graph $G = (V, E)$, 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}.

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

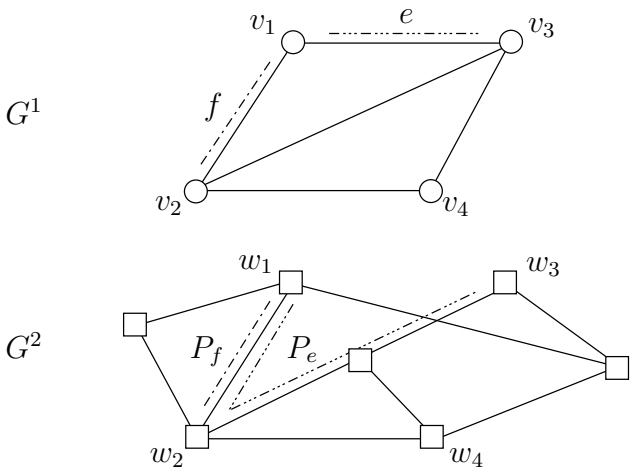


FIG. 4 – Graphs of an IP-over-optical network

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). A chaque sommet $v_i \in V^1$ est associé un sommet $w_i \in V^2$. For an edge $f \in E^1$, we denote by P_f the path in G^2 corresponding to f. Figure 4 shows graphs G^1 and G^2 corresponding to the IP-over-optical network of Figure 1. In G^2, are indicated two paths P_e and P_f which correspond to the edges e and f of G^1.

2.4 Formulations

In terms of graphs, the MCSIPND problem, for both multiple and simple variants, can be presented as follows.

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 $F_e = \{f \in E^1 \,|\, e \in P_f\}$. We let $\mathcal{F} = \{F_e, e \in E^2\}$. Also we denote by D^1 the directed graph associated with G^1 and $\vec{\mathcal{F}} = \{\vec{F_e} \mid F_e \in \mathcal{F}\}$. We denote by K the set of commodities. For each $k \in K$, we know the origin o_k, the destination d_k and the amount ω_k of the demand k.

Let $\mu^1 = 2.5$ Gbit/s and $\mu^2 = 10$ Gbit/s be the possible facilities. For each $ij \in E^1$, let c^l_{ij} be

the cost of installing a capacity μ^l on ij for $l = 1, 2$. Then, the MCSIPND$_m$ problem consists in finding a minimum cost 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 has enough capacity to route the commodities of K with respect to the capacity of the remaining edges.

In what follows we give two different formulations for the MCSIPND problem : the node-arc or conventional formulation and the path or column-generation formulation. A survey of linear multicommodity flow models and solution procedures is presented in [2].

2.4.1 Node-arc formulation

In order to give a node-arc formulation for the MCSIPND problem, let us denote by $f_{uv}^{k,e}$ for an arc $(u, v) \in A^1$, an edge $e \in E^2$ and a commodity $k \in K$, the flow of k on (u, v) from u to v in case of failure of e (*i.e.* when the arcs of $\vec{F}_e$ are removed in D^1). For an edge $uv \in E^1$ let x_{uv}^l be the number of facilities μ^l installed on uv, for $l = 1, 2$. Set

$$b_k^v = \begin{cases} -\omega_k & \text{if } v = o_k, \\ 0 & \text{if } v \neq o_k, d_k, \\ \omega_k & \text{si } v = d_k, \end{cases} \quad \text{for all } v \in V^1, \text{ for all } k \in K.$$

Hence the multiple MCSIPND problem is equivalent to the following integer programming problem.

$$\text{Minimize} \sum_{l=1,2} \sum_{uv \in E^1} c_{uv}^l x_{uv}^l$$

$$\sum_{u:(u,v)\in A^1 \setminus \vec{F}_e} f_{uv}^{k,e} - \sum_{u:(v,u)\in A^1 \setminus \vec{F}_e} f_{vu}^{k,e} = b_k^v \quad \text{for all } v \in V^1, \text{ for all } k \in K, \text{ for all } e \in E^2, \tag{1}$$

$$\sum_{k \in K} f_{uv}^{k,e} \leq \sum_{l=1,2} \mu^l x_{uv}^l \quad \text{for all } uv \in E^1, \text{ for all } e \in E^2, \tag{2}$$

$$\sum_{k \in K} f_{vu}^{k,e} \leq \sum_{l=1,2} \mu^l x_{uv}^l \quad \text{for all } uv \in E^1, \text{ for all } e \in E^2, \tag{3}$$

$$x_{uv}^l \geq 0 \text{ and integer} \quad \text{for all } uv \in E^1, \, l = 1, 2, \tag{4}$$

$$f_{uv}^{k,e}, f_{vu}^{k,e} \geq 0 \quad \text{for all } uv \in E^1, \text{ for all } k \in K, \text{ for all } e \in E^2. \tag{5}$$

Inequalities (1) are called *flow conservation constraints*. Inequalities (2) and (3) express the fact that the sum of the flows of all commodities $k \in K$ on an edge has to be less than or equal to the capacity of this edge. They will be called *capacity constraints*. Inequalities (4) and (5) are called *trivial inequalities*.

By adding the following inequalities

$$x_{uv}^1 + x_{uv}^2 \leq 1 \quad \text{for all } uv \in E^1, \tag{6}$$

and by replacing inequalities (4) by

$$x_{uv}^l \in \{0, 1\} \quad \text{for all } uv \in E^1, \, l = 1, 2, \tag{7}$$

we obtain a valid formulation for the simple MCSIPND problem.

Inequalities (6) express the fact that only one link can be used between two given nodes. Then we have only one type of capacity on an edge. Constraints (7) are the integrality constraints that express the fact that $x_{uv}^l = 1$ if capacity μ^l, $l = 1, 2$, is installed on uv and 0 otherwise.

Note that the following inequalities

$$f^{k,e}_{uv} + f^{k,e}_{vu} \leq \omega_k \quad \text{for all } u, v \in V^1, \text{ for all } k \in K, \text{ for all } e \in E^2, \tag{8}$$

are valid for the MCSIPND problem. They are called *bound inequalities.*

The Arc Residual Capacity inequalities have been introduced in [30] for the Network Loading Problem and used for the Two-Facility capacitated network loading Problem [31]. In the following we extend these inequalities for our problem.

Theorem 2.1 *Let $L \subseteq K$. Set*

$$Q_L = \frac{\sum_{k\in L}\omega_k}{2.5},\ \sigma_L = \left\lceil \frac{Q_L}{4} \right\rceil \text{ and } s_L = \begin{cases} 4 \text{ if } Q_L \bmod 4 = 0, \\ Q_L \bmod 4 \text{ otherwise.} \end{cases}$$

Then we have $Q_L = 4(\sigma_L - 1) + s_L$.
Let $u, v \in V^1$ and $e \in E^2$, then inequality

$$\frac{1}{2.5}\sum_{k\in L}(f^{k,e}_{uv} + f^{k,e}_{vu}) - 2x^1_{uv} - s_L \times 2x^2_{uv} \leq (\sigma_L - 1)(4 - s_L) \tag{9}$$

is valid for the polytope associated with the MCSIPND problem.

Proof. Inequality (9) can be written as

$$\frac{1}{2.5}\sum_{k\in L}(f^{k,e}_{uv} + f^{k,e}_{vu}) \leq Q_L - s_L(\sigma_L - 2x^2_{uv}) + 2x^1_{uv}$$

for $L \subseteq K$, $u, v \in V^1$ and $e \in E^2$ because $(\sigma_L - 1)(4 - s_L) = Q_L - \sigma_L s_L$.

– If $2x^2_{uv} \geq \sigma_L$ then $Q_L - s_L(\sigma_L - 2x^2_{uv}) + 2x^1_{uv} \geq Q_L$.
We know that inequalities (8) are valid for the problem. By summing these inequalities, we obtain

$$\sum_{k\in L}(f^{k,e}_{uv} + f^{k,e}_{vu}) \leq \sum_{k\in L}\omega_k.$$

Then

$$\frac{1}{2.5}\sum_{k\in L}(f^{k,e}_{uv} + f^{k,e}_{vu}) \leq Q_L,$$

which implies that inequality (9) is valid.

– If $2x^2_{uv} \leq \sigma_L - 1$, then
$Q_L - s_L(\sigma_L - 2x^2_{uv}) + 2x^1_{uv}$
$= 4(\sigma_L - 1) + s_L - s_L(\sigma_L - 2x^2_{uv}) + 2x^1_{uv}$
$= 4(\sigma_L - 1) + s_L(2x^2_{uv} - (\sigma_L - 1)) + 2x^1_{uv}$.
Let $s_L = 4 - t$ with $0 \leq t < 4$, we obtain
$Q_L - s_L(\sigma_L - 2x^2_{uv}) + 2x^1_{uv}$
$= 4(\sigma_L - 1) + (4 - t)(2x^2_{uv} - (\sigma_L - 1)) + 2x^1_{uv}$
$= 8x^2_{uv} + 2x^1_{uv} + t(-2x^2_{uv} + (\sigma_L - 1))$.
Then

$$Q_L - s_L(\sigma_L - 2x^2_{uv}) + 2x^1_{uv} \geq 8x^2_{uv} + 2x^1_{uv}, \tag{10}$$

because $t \geq 0$ and $2x^2_{uv} \leq \sigma_L - 1$. As

$$\sum_{k\in K} f^{k,e}_{uv} \leq 2.5x^1_{uv} + 10x^2_{uv} \text{ and } \sum_{k\in K} f^{k,e}_{vu} \leq 2.5x^1_{uv} + 10x^2_{uv},$$

then $\frac{1}{2.5}\sum_{k\in K}(f_{uv}^{k,e}+f_{vu}^{k,e}) \leq 2x_{uv}^1+8x_{uv}^2$, and hence

$$\frac{1}{2.5}\sum_{k\in L}(f_{uv}^{k,e}+f_{vu}^{k,e}) \leq 2x_{uv}^1+8x_{uv}^2. \tag{11}$$

By (10) and (11), it follows that inequality (9) is valid.

□

2.4.2 Path formulation

As in the node-arc formulation, for an edge $uv \in E^1$ we denote by x_{uv}^l the number of facilities μ^l installed on uv for $l = 1, 2$. For an edge $e \in E^2$ and a commodity k we denote by $\mathcal{P}_k^e$ the set of paths from o_k to d_k in the graph $D^1\backslash\vec{F}_e$ (*i.e.* when the edge $e \in E^2$ fails). For a path P of $\mathcal{P}_k^e$, let $y_k^e(P)$ be the amount of flow of commodity k on P in case of faillure of e. Set

$$\tau_{uv}(P) = \begin{cases} 1 & \text{if the arc } (u,v) \text{ belongs to } P, \\ 0 & \text{otherwise,} \end{cases}$$

for all $u, v \in V$. Hence we notice that for all $u, v \in V$, $k \in K$, and $e \in E^2$,

$$f_{uv}^{k,e} = \sum_{P\in\mathcal{P}_k^e} \tau_{uv}(P)y_k^e(P) = \sum_{P\in\mathcal{P}_k^e \,|\, (u,v)\in P} y_k^e(P).$$

By substituting the path variables in the node-arc formulation, we obtain the following mixed integer programming formulation which is valid for the multiple MCSIPND problem.

$$\text{Minimiser} \sum_{l=1,2}\sum_{uv\in E^1} c_{uv}^l x_{uv}^l$$

$$\sum_{P\in\mathcal{P}_k^e} y_k^e(P) = \omega_k \qquad \text{for all } k \in K, \text{ for all } e \in E^2, \tag{12}$$

$$\sum_{k\in K}\sum_{P\in\mathcal{P}_k^e \,|\, (u,v)\in P} y_k^e(P) \leq \sum_{l=1,2} \mu^l x_{uv}^l \qquad \text{for all } uv \in E^1, \text{ for all } e \in E^2, \tag{13}$$

$$\sum_{k\in K}\sum_{P\in\mathcal{P}_k^e \,|\, (v,u)\in P} y_k^e(P) \leq \sum_{l=1,2} \mu^l x_{uv}^l \qquad \text{for all } uv \in E^1, \text{ for all } e \in E^2, \tag{14}$$

$$y_k^e(P) \geq 0 \qquad \text{for all } e \in E^2, \text{ for all } k \in K, \text{ for all } P \in \mathcal{P}_k^e, \tag{15}$$

$$x_{uv}^l \geq 0 \text{ and integer} \qquad \text{for all } uv \in E^1,\ l = 1, 2. \tag{16}$$

This formulation has a collection of $|K|$ demand constraints (12) that represent the flow of each path P in $\mathcal{P}_k^e$, $k \in K$ for each failure $e \in E^2$ and ($\sum_{P\in\mathcal{P}_k^e} y_k^e(P)$ represents the amount of flow of commodity k passing through the set of paths from o_k to d_k). This flow has to be equal to the amount ω_k between o_k and d_k. Inequalities (13) and (14) are called *capacity constraints*. The flow through the edge uv has to be less than the capacity of this edge from u to v (constraints (13)) and from v to u (constraints (14)). Inequalities (15) and (16) are the *trivial constraints*.

By adding inequalities (6) and replacing inequalities (16) by inequalities (7), we obtain a valid formulation for the simple MCSIPND problem.

The linear relaxation of the node-arc MCSIPND formulation contains a large number of constraints and a large number of variables. The linear relaxation of the path formulation, however, contains a moderate number of constraints (for each failure, one for each commodity and one for each arc) and a huge number of variables (one for each path for each commodity for each failure). These linear relaxations may require excessive memory and times to solve. An appropriate method to solve this second type of formulation would be the column generation approach. In the next section, we discuss this approach.

3 Column generation

Column generation will be used to solve the linear relaxation of the path based formulation of the MCSIPND problem (called the *master problem*). This approach has been extensively used for modeling and solving large versions of the linear multicommodity flow problem [2, 5, 29]. The general idea of column generation is to solve a restricted linear program with a small number of columns (variables) in order to determine an optimal solution for the master problem. In fact a limited number of variables may induce an optimal basis solution for the master problem. So the column generation algorithm solves the linear relaxation of the master problem by solving the linear relaxations of several restricted master problems. After determining the solution of the linear relaxation of a restricted master problem, we use the *pricing problem* which consists in finding whether there are any columns not yet in the restricted master problem with negative red uced cost. If none can be found, the current solution is optimal for the linear relaxation of the master problem. However, if one or more such columns do exist, then they are added to the restricted master problem and the process is repeated. This approach could be combined with row generation to obtain a very strong method to solve the linear relaxations (see [6]).

3.1 Initial Solution

To start the column generation scheme, an initial restricted master problem has to be provided. This initial problem must have a feasible solution to ensure that correct information is passed to the pricing problem.

For the version of the MCSIPND problem with multiple edges (MCSIPND$_m$ problem), finding an initial feasible solution is very easy. Indeed we look for shortest paths between the origin-destinations of all commodities. These paths are then used to carry the flow for each commodity. As we can install as much capacity as we want, this multicommodity flow is feasible.

For the MCSIPND$_s$ problem, we consider the following linear program obtained from the path based formulation by setting $x^1_{uv} = 0$ and $x^2_{uv} = 1$ for all edge $uv \in E^1$, that is to say by fiwing a capacity of 10 Gbits for each edge. We also consider a new variable ε.

Minimiser ε

$$\sum_{P \in \mathcal{P}^e_k} y^e_k(P) = \omega_k \qquad \text{for all } k \in K, \text{ for all } e \in E^2, \tag{17}$$

$$\sum_{k \in K} \sum_{P \in \mathcal{P}^e_k \,|\, (u,v) \in P} y^e_k(P) \leq \mu^2 + \varepsilon \qquad \text{for all } (u,v) \in A^1, \text{ for all } e \in E^2, \tag{18}$$

$$y^e_k(P) \geq 0 \qquad \text{for all } e \in E^2, \text{ for all } k \in K, \text{ for all } P \in \mathcal{P}^e_k, \tag{19}$$

$$\varepsilon \geq 0. \tag{20}$$

At the optimum, variable ε corresponds to the minimum amount of capacity we must add to each edge in order to allow a multicommodity flow. We solve this linear program using a column generation algorithm similar to the algorithm used to solve the MCSIPND$_m$ problem. If the optimal solution for this linear program imposes that $\varepsilon > 0$, we conclude that the MCSIPND$_s$ problem has no solution. On the other hand, if $\varepsilon = 0$, the set of variables used in the column generation permits to have an initial feasible solution for the restricted master problem.

When defining the initial restricted master problem, it is necessary to ensure the existence of a feasible solution. Finding a "good" initial restricted master problem could be important. Indeed this would permit to determine the initial dual variables which will be passed to the pricing problem.

3.2 Pricing problem

For any restricted master problem, let γ_k^e, $\vartheta^e_{(u,v)}$ and $\vartheta^e_{(v,u)}$ be the dual variables associated with constraints (12), (13) and (14), respectively. The reduced cost associated with the variable of a path $P \in \mathcal{P}_k^e$ is $R_P^{k,e} = \sum_{(u,v)\in P} \vartheta^e_{(u,v)} - \gamma_k^e$. The pricing problem can then be reduced to the search of several shortest path problem with non-negative costs. Indeed the princing problem consist in finding for each commodity $k \in K$ and each edge $e \in E^2$, a path P in $\mathcal{P}_k^e$ such that $R_P^{k,e} = \min_{P'\in\mathcal{P}_k^e} R_{P'}^{k,e}$ and $R_P^{k,e} < 0$. Therefore, we can identify columns which have to be added to the restricted master problem by solving one shortest path problem for each commodity $k \in K$ and each edge $e \in E^2$ in the graph with arc costs equal to $\vartheta^e_{(u,v)}$ for each $(u,v) \in A^1\backslash\vec{F}_e$. If one or more paths have non-positive reduced cost, then they are added in the restricted master problem. Otherwise, the master problem has been solved to optimality.

Combining column and row generation can yield a very strong linear relaxation. In the next section, we describe some valid inequalities. These will be used as cutting planes in our Branch-and-Cut-and-Price algorithm for the two variants of the problem. We introduce some inequalities which are valid for the problem with or without multiple edges.

4 Valid inequalities and facets

Throughout the following sections we consider a graph $G = (V, E)$ and the associated digraph $D = (V, A)$ obtained from G by substituting each edge of E by two arcs. We consider also a family $\mathcal{F} = \{F_1, \ldots, F_t\} \subseteq 2^E$, $t \geq 2$ of edge subsets of E and the family $\vec{\mathcal{F}} = \{\vec{F}_1, \ldots, \vec{F}_t\} \subseteq 2^A$ of arc subsets associated with $\mathcal{F}$. Let K be a set of demands. For an arc $(u,v) \in A$, a commodity $k \in K$ and $i \in \{1, \ldots, t\}$, let us denote by $f_{uv}^{k,i}$ the flow of k on (u,v) from u to v when the arcs of $\vec{F}_i$ are removed in D^1. For $i \in \{1, \ldots, t\}$, we will denote by $G_i = (V, E_i)$ (resp. $D_i = (V, A_i)$) the subgraph of G (resp. D) obtained by removing the edges of F_i (resp. $\vec{F}_i$). Hence $E_i = E\backslash F_i$ (resp. $A_i = A\backslash\vec{F}_i$). In the following we consider $\mu^1 = 2.5$ and $\mu^2 = 10$.

Now, consider the following inequalities :

$$\sum_{u:(u,v)\in A\backslash\vec{F}_i} f_{uv}^{k,i} - \sum_{u:(v,u)\in A\backslash\vec{F}_i} f_{vu}^{k,i} = b_k^v \qquad \text{for all } v \in V, \text{ for all } k \in K,\ i = 1, \ldots, t, \tag{21}$$

$$\sum_{k\in K} f_{uv}^{k,i} \leq 2.5x_{uv}^1 + 10x_{uv}^2 \qquad \text{for all } uv \in E,\ i = 1, \ldots, t, \tag{22}$$

$$\sum_{k\in K} f_{vu}^{k,i} \leq 2.5x_{uv}^1 + 10x_{uv}^2 \qquad \text{for all } uv \in E,\ i = 1, \ldots, t, \tag{23}$$

$$x_{uv}^l \geq 0 \qquad \text{for all } uv \in E,\ l = 1, 2, \tag{24}$$

$$f_{uv}^{k,i}, f_{vu}^{k,i} \geq 0 \qquad \text{for all } uv \in E, \text{for all } k \in K, i = 1, \ldots, t, \tag{25}$$

$$x_{uv}^1 + x_{uv}^2 \leq 1 \qquad \text{for all } uv \in E. \tag{26}$$

Let $\text{MCSIPND}_m^{na}(G, \mathcal{F}, K)$ and $\text{MCSIPND}_s^{na}(G, \mathcal{F}, K)$ be the polytopes associated with the MCSIPND_m problem and the MCSIPND_s problem (when $G = G^1$ and $\mathcal{F} = \{F_e,\ e \in E^2\}$), *i.e.*

$$\text{MCSIPND}_m^{na}(G, \mathcal{F}, K) = \{(x, f) \in \mathcal{D}_m^{na} \mid x \text{ and } f \text{ satisfy (21)-(25)}\},$$

$$\text{MCSIPND}_s^{na}(G, \mathcal{F}, K) = \{(x, f) \in \mathcal{D}_s^{na} \mid x \text{ and } f \text{ satisfy (21)-(26)}\}$$

with

$$\mathcal{D}_m^{na} = \{x \in I\!N^{2|E|}, f \in I\!R^{2|E|\times|K|\times|\mathcal{F}|}\}$$

and

$$\mathcal{D}_s^{na} = \{x \in \{0,1\}^{2|E|}, f \in I\!R^{2|E|\times|K|\times|\mathcal{F}|}\}.$$

Let $\mathcal{P}_k^i$ be the set of paths between o_k and d_k in the graph $D\backslash\vec{F}_i$ for $i \in \{1,\ldots,t\}$. Let $\text{MCSIPND}_m^p(G,\mathcal{F},K)$ denote the convex hull of the integer solutions of the system

$$\sum_{P\in\mathcal{P}_k^i} y_k^i(P) = \omega_k \qquad \text{for all } k\in K,\ i=1,\ldots,t, \tag{27}$$

$$\sum_{k\in K}\sum_{P\in\mathcal{P}_k^i \mid (u,v)\in P} y_k^i(P) \le 2.5x_{uv}^1 + 10x_{uv}^2 \qquad \text{for all } uv\in E,\ i=1,\ldots,t, \tag{28}$$

$$\sum_{k\in K}\sum_{P\in\mathcal{P}_k^i \mid (v,u)\in P} y_k^i(P) \le 2.5x_{uv}^1 + 10x_{uv}^2 \qquad \text{for all } uv\in E,\ i=1,\ldots,t, \tag{29}$$

$$x_{uv}^l \ge 0 \qquad \text{for all } uv\in E,\ l=1,2, \tag{30}$$

$$y_k^i(P) \ge 0 \qquad \text{for all } k\in K,\ i=1,\ldots,t, \text{ for all } P\in\mathcal{P}_k^i. \tag{31}$$

By adding constraints (26) to $\text{MCSIPND}_m^p(G,\mathcal{F},K)$, we obtained $\text{MCSIPND}_s^p(G,\mathcal{F},K)$.

We can remark that if $G = G^1$ and $\mathcal{F} = \{F_e, e\in E^2\}$, $\text{MCSIPND}_m^p(G,\mathcal{F},K)$ (resp. $\text{MCSIPND}_s^p(G,\mathcal{F},K)$) is nothing but the polytope associated with the MCSIPND_m (resp. MCSIPND_s) problem.

If no confusion may arise, we will sometimes write $\text{MCSIPND}_m(G,\mathcal{F},K)$ for the two polytopes $\text{MCSIPND}_m^{na}(G,\mathcal{F},K)$ and $\text{MCSIPND}_m^p(G,\mathcal{F},K)$. Similarly the polytope $\text{MCSIPND}_s(G,\mathcal{F},K)$ will correspond indifferently to the polytopes $\text{MCSIPND}_s^{na}(G,\mathcal{F},K)$ and $\text{MCSIPND}_s^p(G,\mathcal{F},K)$.

The following theorem gives the dimension of the polytope $\text{MCSIPND}_m^{na}(G,\mathcal{F},K)$.

Theorem 4.1 *$dim(MCSIPND_m^{na}(G,\mathcal{F},K)) = 2|E| + 2|\mathcal{F}|\times|K| - (|N|-1)|\mathcal{F}|\times|K|$.*

Proof. The node-arc formulation of the MCSIPND_m problem contains $2|E| + 2|\mathcal{F}|\times|E|\times|K|$ variables and $(|N|-1)|\mathcal{F}|\times|K|$ nonredondant equality constraints. Hence $dim(MCSIPND_m^{na}(G,\mathcal{F},K)) \le 2|E| + 2|\mathcal{F}|\times|E|\times|K| - (|N|-1)|\mathcal{F}|\times|K|$.
The proof of $dim(MCSIPND_m^{na}(G,\mathcal{F},K)) \ge 2|E| + 2|\mathcal{F}|\times|E|\times|K| - (|N|-1)|\mathcal{F}|\times|K|$ uses arguments similar to those used in theorem 4.9 and we, therefore, omit it. □

In the following, we introduce several classes of valid inequalities. We also give necessary conditions and sufficient conditions for one of these inequalities to be facet defining. We assume that the reader is familiar with polyhedral combinatorics, for more details see [40].

As in [7], 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.

4.1 Cut inequalities

For $W\subseteq V$, we denote by $\gamma^+(W)\subseteq K$ (resp. $\gamma^-(W)\subseteq K$) the set of demands which have their origin (resp. destination) in W and their destination (resp. origin) in $V\backslash W$. We denote also by $\gamma(W)$ the set $\gamma^+(W)\cup\gamma^-(W)$.

Theorem 4.2 *Let $F_i\in\mathcal{F}$ be an edge subset of E and $W\subseteq V$, $\emptyset\neq W\neq V$ such that $\gamma(W)\neq\emptyset$. Then the inequality*

$$x^1(\delta_{G_i}(W)) + x^2(\delta_{G_i}(W)) \ge 1 \tag{32}$$

is valid for $MCSIPND_m(G,\mathcal{F},K)$ and $MCSIPND_s(G,\mathcal{F},K)$.

Proof. Trivial. □

Inequalities of type (32) will be called *design cut inequalities*. Theses inequalities express the fact that the graph G keeps connectivity between the origin and the destination of each demand after removing the edges of F_i, for all $F_i \in \mathcal{F}$.

Given a set of nodes $W \subseteq V$, $\emptyset \neq W \neq V$, let

$$D_W = \left\lceil \max \left\{ \frac{\sum_{k\in\gamma^+(W)} \omega_k}{2.5}, \frac{\sum_{k\in\gamma^-(W)} \omega_k}{2.5} \right\} \right\rceil.$$

Let $F_i \in \mathcal{F}$ be an edge subset of E and $W \subseteq V$, $\emptyset \neq W \neq V$. Consider the inequality

$$x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) \geq D_W. \tag{33}$$

We have the following.

Theorem 4.3 *Inequality (33) is valid for both MCSIPND$_m$$(G, \mathcal{F}, K)$ and MCSIPND$_s$$(G, \mathcal{F}, K)$.*

Proof. The aggregate capacity across $\delta_{G_i}(W)$ must be no less than the demand across the cut from W to $V\backslash W$ and from $V\backslash W$ to W. Thus one should have

$$2.5x^1(\delta_{G_i}(W)) + 10x^2(\delta_{G_i}(W)) \geq \max \left\{ \sum_{k\in\gamma^+(W)} \omega_k, \sum_{k\in\gamma^-(W)} \omega_k \right\}.$$

Dividing by 2.5 and rounding up the right hand side yields (33). □

Before introducing the next class of inequalities, we give a lemma.

Lemma 4.4 *Let $F_i \in \mathcal{F}$ be an edge subset of E and $W \subseteq V$, $\emptyset \neq W \neq V$. Then inequality*

$$x^1(\delta_{G_i}(W)) + 2x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{D_W}{2} \right\rceil \tag{34}$$

is valid for both MCSIPND$_m$$(G, \mathcal{F}, K)$ and MCSIPND$_s$$(G, \mathcal{F}, K)$.

Proof. The following inequalities are valid for both MCSIPND$_m(G, \mathcal{F}, K)$ and MCSIPND$_s(G, \mathcal{F}, K)$,

$$\begin{aligned} x^1(\delta_{G_i}(W) + 4x^2(\delta_{G_i}(W)) &\geq D_W, \\ x^1_{uv} &\geq 0 \quad \text{for all } uv \in \delta_{G_i}(W). \end{aligned}$$

By summing these inequalities, we obtain

$$2x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) \geq D_W.$$

Dividing by 2 and rounding up the right hand side yields inequality (34). □

Theorem 4.5 *Let $F_i \in \mathcal{F}$ be an edge subset of E and $W \subseteq V$, $\emptyset \neq W \neq V$. Then inequality*

$$x^1(\delta_{G_i}(W)) + x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{D_W}{4} \right\rceil \tag{35}$$

is valid for both MCSIPND$_m$$(G, \mathcal{F}, K)$ and MCSIPND$_s$$(G, \mathcal{F}, K)$.

Proof. By adding the inequalities

$$x^1_{uv} \geq 0 \qquad \text{for all } uv \in \delta_{G_i}(W).$$

to inequality (34), we get

$$2x^1(\delta_{G_i}(W)) + 2x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{D_W}{2} \right\rceil.$$

Dividing by 2 and rounding up the right hand side yields

$$x^1(\delta_{G_i}(W)) + x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{\left\lceil \frac{D_W}{2} \right\rceil}{2} \right\rceil.$$

As $\left\lceil \frac{\left\lceil \frac{D_W}{2} \right\rceil}{2} \right\rceil = \left\lceil \frac{D_W}{4} \right\rceil$, we get inequality (35). □

Theorem 4.6 *Let $F_i \in \mathcal{F}$ be an edge subset of E and $W \subseteq V$, $\emptyset \neq W \neq V$. Then inequality*

$$x^1(\delta_{G_i}(W)) + 3x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{3D_W}{4} \right\rceil \tag{36}$$

is valid for both MCSIPND$_m$$(G, \mathcal{F}, K)$ and MCSIPND$_s$$(G, \mathcal{F}, K)$.

Proof. The following inequalities are valid for both MCSIPND$_m(G, \mathcal{F}, K)$ and MCSIPND$_s(G, \mathcal{F}, K)$,

$$\begin{aligned} x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) &\geq D_W \\ x^1(\delta_{G_i}(W)) + 2x^2(\delta_{G_i}(W)) &\geq \left\lceil \frac{D_W}{2} \right\rceil \end{aligned}$$

By summing these inequalities we obtain

$$2x^1(\delta_{G_i}(W)) + 6x^2(\delta_{G_i}(W)) \geq D_W + \left\lceil \frac{D_W}{2} \right\rceil.$$

Dividing by 2 and rounding up the right hand side yields

$$x^1(\delta_{G_i}(W)) + 3x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{\left\lceil \frac{3D_W}{2} \right\rceil}{2} \right\rceil.$$

As $\left\lceil \frac{\left\lceil \frac{3D_W}{2} \right\rceil}{2} \right\rceil = \left\lceil \frac{3D_W}{4} \right\rceil$, we get inequality (36). □

Theorem 4.7 *Let $F_i \in \mathcal{F}$ be an edge subset of E and $W \subseteq V$, $\emptyset \neq W \neq V$. Then inequality*

$$x^1(\delta_{G_i}(W)) + 2x^2(\delta_{G_i}(W)) \geq \begin{cases} \left\lceil \frac{D_W}{2} \right\rceil + 1 & \text{if } D_W \bmod 4 = 2, \\ \left\lceil \frac{D_W}{2} \right\rceil & \text{otherwise}, \end{cases} \tag{37}$$

is valid for both MCSIPND$_m$$(G, \mathcal{F}, K)$ and MCSIPND$_s$$(G, \mathcal{F}, K)$.

Proof. The following inequalities are valid for both $\mathrm{MCSIPND}_m(G,\mathcal{F},K)$ and $\mathrm{MCSIPND}_s(G,\mathcal{F},K)$,

$$\begin{aligned} x^1(\delta_{G_i}(W)) + x^2(\delta_{G_i}(W)) &\geq \left\lceil \frac{D_W}{4} \right\rceil, \\ x^1(\delta_{G_i}(W)) + 3x^2(\delta_{G_i}(W)) &\geq \left\lceil \frac{3D_W}{4} \right\rceil. \end{aligned}$$

By summing these inequalities we obtain

$$2x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{D_W}{4} \right\rceil + \left\lceil \frac{3D_W}{4} \right\rceil.$$

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

$$x^1(\delta_{G_i}(W)) + 2x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{\left\lceil \frac{3D_W}{4} \right\rceil + \left\lceil \frac{D_W}{4} \right\rceil}{2} \right\rceil.$$

As $\left\lceil \frac{\left\lceil \frac{3D_W}{4} \right\rceil + \left\lceil \frac{D_W}{4} \right\rceil}{2} \right\rceil = \begin{cases} \left\lceil \frac{D_W}{2} \right\rceil + 1 & \text{if } D_W \bmod 4 = 2, \\ \left\lceil \frac{D_W}{2} \right\rceil & \text{otherwise}, \end{cases}$

the theorem follows. □

Inequalities (33), (35), (36), (37) will be called *capacity demand cut inequalities.*

One may generate further cut based valid inequalities by combining inequalities of type (33), (35), (36), (37) and trivial inequalities. However all inequalities obtained this way are redundant with respect to the capacity demand cut inequalities (see [8]).

In the following, we give necessary conditions and sufficient conditions for inequality (33) to be facet defining for $\mathrm{MCSIPND}_m^{na}(G,\mathcal{F},K)$.

Theorem 4.8 *Inequality (33) defines a facet of $MCSIPND_m^{na}(G,\mathcal{F},K)$ only if*

1. *$G_i(W)$ and $G_i(\overline{W})$ are connected,*
2. *there is no $j \in \{1,\ldots,t\}\backslash\{i\}$ such that $F_i \cap \delta_G(W) \subset F_j \cap \delta_G(W)$,*
3. *$G(W)$ and $G(\overline{W})$ are $\mathcal{F}$-connected, if $\delta_G(W) \cap F_i = \emptyset$,*
4. *$D_W > max\left\{ \frac{\sum_{k\in\gamma^+(W)} \omega_k}{2.5}, \frac{\sum_{k\in\gamma^-(W)} \omega_k}{2.5} \right\}$,*
5. *$D_W \geq 4$.*

Proof. 1. Suppose w.l.o.g., that $G_i(W)$ is not connected. Hence there is a partition W_1, W_2 of W such that $\delta_{G_i}(W_1,W_2) = \emptyset$ (see Fig. 5). Thus $\delta_{G_i}(W_1) = \delta_{G_i}(W_1,\overline{W})$ and $\delta_{G_i}(W_2) = \delta_{G_i}(W_2,\overline{W})$.

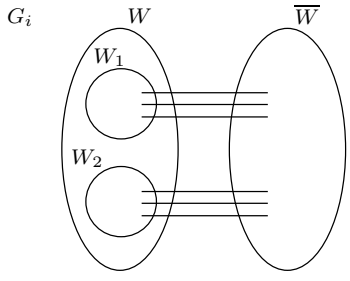


FIG. 5 –

This implies that

$$\begin{aligned} & x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) \\ & \quad = \; x^1(\delta_{G_i}(W_1, \overline{W})) + 4x^2(\delta_{G_i}(W_1, \overline{W})) + x^1(\delta_{G_i}(W_2, \overline{W})) + 4x^2(\delta_{G_i}(W_2, \overline{W})) \\ & \quad = \; x^1(\delta_{G_i}(W_1)) + 4x^2(\delta_{G_i}(W_1)) + x^1(\delta_{G_i}(W_2)) + 4x^2(\delta_{G_i}(W_2)) \\ & \quad \geq \; D_{W_1} + D_{W_2}. \end{aligned}$$

The last inequality is obtain from the inequalities of type (33) corresponding to W_1 and W_2. As $D_{W_1} + D_{W_2} \geq D_W$, inequality (33) is then redundant with respect to these inequalities and hence (33) can not be facet defining.

2. Assume the contrary. Let $j \in \{1, \ldots, t\} \backslash \{i\}$ such that $F_i \cap \delta_G(W) \subset F_j \cap \delta_G(W)$. Then (33) can be obtained as the sum of the following valid constraints.

$$\begin{aligned} x^1(\delta_{G_j}(W)) + 4x^2(\delta_{G_j}(W)) & \geq D_W, \\ x^1_{uv} & \geq 0 \qquad \text{for all } uv \in (F_j \backslash F_i) \cap \delta_G(W), \\ 4x^2_{uv} & \geq 0 \qquad \text{for all } uv \in (F_j \backslash F_i) \cap \delta_G(W). \end{aligned}$$

Hence it is not facet defining.

3. Suppose $\delta_G(W) \cap F_i = \emptyset$ and w.l.o.g., that $G(W)$ is not $\mathcal{F}$-connected. Then there is $j \in \{1, \ldots, t\}$ such that $G_j(W)$ is not connected. If $j = i$, then by condition 1), (33) can not define a facet. So suppose $j \neq i$. If $\delta_G(W) \cap F_j \neq \emptyset$, then by condition 2), (33) can not also define a facet. Then, suppose that $\delta_G(W) \cap F_j = \emptyset$. Hence, $\delta_G(W) = \delta_{G_i}(W) = \delta_{G_j}(W)$. As $G_j(W)$ is not connected, there is a partition W_1, W_2 of W such that $\delta_{G_j}(W_1, W_2) = \emptyset$. Hence $\delta_{G_j}(W_1) = \delta_{G_j}(W_1, \overline{W})$ and $\delta_{G_j}(W_2) = \delta_{G_j}(W_2, \overline{W})$. This implies that

$$\begin{aligned} & x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) \\ & \quad = \; x^1(\delta_{G_j}(W)) + 4x^2(\delta_{G_j}(W)) \\ & \quad = \; x^1(\delta_{G_j}(W_1, \overline{W})) + 4x^2(\delta_{G_j}(W_1, \overline{W})) + x^1(\delta_{G_j}(W_2, \overline{W})) + 4x^2(\delta_{G_j}(W_2, \overline{W})) \\ & \quad = \; x^1(\delta_{G_j}(W_1)) + 4x^2(\delta_{G_j}(W_1)) + x^1(\delta_{G_j}(W_2)) + 4x^2(\delta_{G_j}(W_2)) \\ & \quad \geq \; D_{W_1} + D_{W_2}. \end{aligned}$$

Since $D_{W_1} + D_{W_2} \geq D_W$, as before, inequality (33) cannot then define a facet.

4. If $D_W = \max \left\{ \frac{\sum_{k \in \gamma^+(W)} \omega_k}{2.5}, \frac{\sum_{k \in \gamma^-(W)} \omega_k}{2.5} \right\}$, then by Proposition **??** constraint (33) is redondant with respect constraints (21)-(23), and do not then facet defining.

5. If $D_W < 4$, then for every solution of MCSIPND^{na}_m that satisfies (33) with equality we have $x^2(e) = 0$ for all $e \in \delta_{G_i}(W)$.

   If $|\delta_{G_i}(W)| \geq 2$, then there are two edges $e_1, e_2 \in \delta_{G_i}(W)$ such that $x^2(e_1) = x^2(e_2) = 0$ in every solution satisfying (33) with equality. But in this case, we can not construct $\dim(\text{MCSIPND}^{na}_s(G, \mathcal{F}, K))$ solutions satisfying (33) with equality, and affinely independant. This implies that (33) can not be facet defining.

   So suppose that $\delta_{G_i}(W) = \{e = uv\}$ with $u \in W$ and $v \in \overline{W}$. Then in every solution satisfying (33) with equality, one should have

$$\begin{aligned} x^1(e) & = D_W, \\ x^2(e) & = 0. \end{aligned}$$

Also, in every solution satisfying (33) with equality, one should have

$$f^{k,i}_{uv} \; = \; \begin{cases} \omega_k + \varepsilon^k \text{ if } k \in \gamma^+(W) \\ \varepsilon^k \text{ if } k \in \gamma^-(W) \end{cases}$$

$$f_{vu}^{k,i} = \begin{cases} \omega_k + \varepsilon^k \text{ if } k \in \gamma^-(W) \\ \varepsilon^k \text{ if } k \in \gamma^+(W) \end{cases}$$

where ε^k, $k \in \gamma(W)$ are positive and sufficiently small. In fact, we can see that the variables f are not involved in constraint (33). Furthermore, as by condition 4), $D_W > \max\left\{\frac{\sum_{k\in\gamma^+(W)}\omega_k}{2.5}, \frac{\sum_{k\in\gamma^-(W)}\omega_k}{2.5}\right\}$, the available capacity is yet more than the demand across the cut. This gives the possibility to route a lightly upper flow. This explains the addition of ε^k from u to v and from v to u.

Set the matrix whose columns correspond to solutions satisfying (33) with equality and the lines are associated to $x^1(e)$, $x^2(e)$, the sum of variables $f_{uv}^{k,i}$ for $k \in \gamma(W)$ and the sum of variables $f_{vu}^{k,i}$ for $k \in \gamma(W)$. We denote by p the number of solutions satisfying (33) with equality. This matrix has the following form :

$$\begin{array}{l} x^1(e) \\ x^2(e) \\ \sum\limits_{k\in\gamma(W)} f_{uv}^{k,i} \\ \sum\limits_{k\in\gamma(W)} f_{vu}^{k,i} \end{array} \begin{pmatrix} D_W & \cdots & D_W \\ 0 & \cdots & 0 \\ \sum\limits_{k\in\gamma^+(W)} \omega_k + \sum\limits_{k\in\gamma(W)} \varepsilon_1^k & \cdots & \sum\limits_{k\in\gamma^+(W)} \omega_k + \sum\limits_{k\in\gamma(W)} \varepsilon_p^k \\ \sum\limits_{k\in\gamma^-(W)} \omega_k + \sum\limits_{k\in\gamma(W)} \varepsilon_1^k & \cdots & \sum\limits_{k\in\gamma^-(W)} \omega_k + \sum\limits_{k\in\gamma(W)} \varepsilon_p^k \end{pmatrix}.$$

By substracting the fourth line from the third one, we get a line which is multiple of the first one. As the second line is formed only by zeros, we can not get as many as necessary of solution affinelly independant for (33) to be facet defining.

□

Theorem 4.9 *Inequality (33) defines a facet of $MCSIPND_m^{na}(G, \mathcal{F}, K)$ if*

1. *condition 1), 2), 4), 5) of Theorem 4.8 are satisfied,*
2. *$G(W)$ and $G(\overline{W})$ are $\mathcal{F}$-connected.*

Proof. See Appendix. □

In [31], Magnanti et al. introduce cutset inequalities valid for the Two-Facility Capacitated Network Loading Problem (TFLP). These can be easily extended to the $MCSIPND_m$ and the $MCSIPND_s$ problems. The extended ones are special cases of the capacity demand cut inequalities.

Corollary 4.10 *Let $F_i \in \mathcal{F}$ be an edge subset of E, $W \subseteq V$, $\emptyset \neq W \neq V$, and let*

$$r_W = \begin{cases} 4 \text{ if } D_W \bmod 4 = 0, \\ D_W \bmod 4 \text{ otherwise.} \end{cases}$$

Then the inequality

$$\sum_{e\in\delta_{G_i}(W)} (x_e^1 + r_W x_e^2) \geq r_W \left\lceil \frac{D_W}{4} \right\rceil \tag{38}$$

is valid for both $MCSIPND_m(G, \mathcal{F}, K)$ and $MCSIPND_s(G, \mathcal{F}, K)$.

Proof.
- If $r_W = 1$, inequality (38) is nothing but inequality (35).
- If $r_W = 2$, as $2\left\lceil \frac{D_W}{4} \right\rceil = \left\lceil \frac{D_W}{2} \right\rceil + 1$, inequality (38) is nothing but inequality (37).
- If $r_W = 3$, as $3\left\lceil \frac{D_W}{4} \right\rceil = \left\lceil \frac{3D_W}{4} \right\rceil$, inequality (38) is nothing but inequality (36).
- If $r_W = 4$, as $4\left\lceil \frac{D_W}{4} \right\rceil = D_W$, inequality (38) is nothing but inequality (33).

□

4.2 Saturation inequalities

In this section we introduce a further class of valid inequalities for only the polytope $\text{MCSIPND}_s(G, \mathcal{F}, K)$. These inequalities are also induced by cuts. In these inequalities only the x^2 variables, associated with the links with big capacity (10Gbits), are involved. In fact, the idea behind these inequalities is that if the demand can not be routed on the links of a cut with small capacity (2.5 Gbits), then at least one big capacity must be installed on one of the links of the cut.

Theorem 4.11 *Let $F_i \in \mathcal{F}$ and $W \subseteq V$, $\emptyset \neq W \neq V$. Then inequality*

$$x^2(\delta_{G_i}(W)) \geq \left\lceil \frac{max\ \left\{\sum_{k\in\gamma^+(W)} \omega_k, \sum_{k\in\gamma^-(W)} \omega_k\right\} - |\delta_{G_i}(W)| \times 2.5}{7.5} \right\rceil \tag{39}$$

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

Proof. We suppose that $x^1(e) = 1$ for all $e \in \delta_{G_i}(W)$. The available capacity across the cut $\delta_{G_i}(W)$ is then equal to $|\delta_{G_i}(W)| \times 2.5$. Let $M = \max\ \left\{\sum_{k\in\gamma^+(W)} \omega_k, \sum_{k\in\gamma^-(W)} \omega_k\right\} - |\delta_{G_i}(W)| \times 2.5$. Note that M is nothing but the missing capacity across the cut $\delta_{G_i}(W)$.

- If $M \leq 0$, that is to say, the demand over the cut is less than or equal the available capacity on the small edges. Then the edges can carry the flow across $\delta_{G_i}(W)$. In this case, inequality (39) can be written as $x^2(\delta_{G_i}(W)) \geq 0$ and is thus valid for $\text{MCSIPND}_s(G, \mathcal{F}, K)$.
- If $M > 0$, this means that the edges of $\delta_{G_i}(W)$ do not suffice to carry the whole flow between W and $\overline{W}$, if they get all a small capacity of 2.5 Gbits. We then have to replace some small capacities by big ones. Let $f \in E^1$ be an edge on which we install a big capacity instead of a small one. As $x^1(e) + x^2(e) \leq 1$ for all $e \in E^1$, if $x^2(f) = 1$ this implies that $x^1(f) = 0$. Therefore, installing a big capacity on f allows to add a capacity of 10-2.5=7.5 Gbits over the cut. Hence the minimum number of edges with big capacity we need in order to route the demand between W and $\overline{W}$ is $\left\lceil \frac{\max\ \left\{\sum_{k\in\gamma^+(W)} \omega_k, \sum_{k\in\gamma^-(W)} \omega_k\right\} - |\delta_{G_i}(W)| \times 2.5}{7.5} \right\rceil$. Then inequality (39) is valid for $\text{MCSIPND}_s(G, \mathcal{F}, K)$.

□

In the two following sections we present further classes of valid inequalities which are extensions of valid inequalities introduced in [7] for the problem without capacities.

4.3 Cut-cycle inequalities

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 $T_2 = \delta_G(W) \backslash (T_1 \cup (\bigcap_{i=1,\ldots,s} F_{j_i}))$. Such a configuration (W, T_1, T_2) will be called a *cut-cycle configuration* (see Figure 6).

Theorem 4.12 *Let (W, T_1, T_2) be a cut-cycle configuration such that $\gamma(W) \neq \emptyset$. For $e \in \delta_G(W)$, let $r_e = |\{i \in \{1, \ldots, s\} \,|\, e \in \delta_G(W) \backslash F_{j_i}\}|$ and r be the smallest integer such that $r(s - q) \geq \max_{e\in T_2}\{r_e\}$. Then inequality*

$$\sum_{l=1,2} (x^l(T_1) + r x^l(T_2)) \geq \left\lceil \frac{s}{s-q} \right\rceil \tag{40}$$

is valid for both $MCSIPND_m(G, \mathcal{F}, K)$ and $MCSIPND_s(G, \mathcal{F}, K)$.

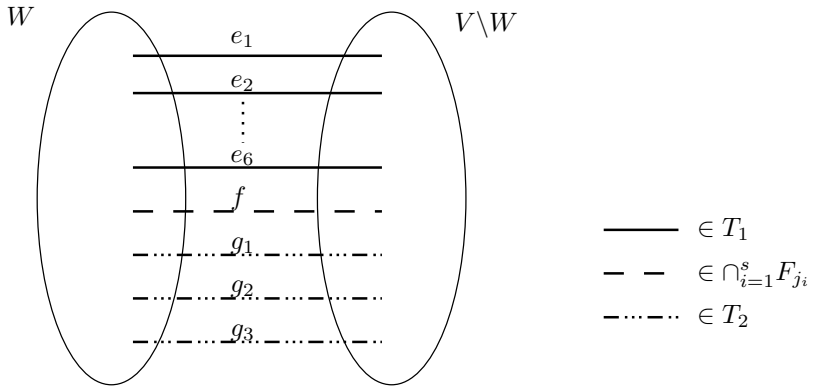


FIG. 6 – A cut-cycle configuration

Proof. The following inequalities are valid for both $\text{MCSIPND}_m(G, \mathcal{F}, K)$ and $\text{MCSIPND}_s(G, \mathcal{F}, K)$,

$$\sum_{l=1,2} x^l(\delta_{G_{j_i}}(W)) \geq 1 \qquad \text{for } i = 1, \ldots, s,$$
$$(r(s-q) - r_e)x^l(e) \geq 0 \qquad \text{for all } e \in T_2, l = 1, 2.$$

By summing these inequalities, we obtain

$$\sum_{l=1,2} ((s-q)x^l(T_1) + r(s-q)x^l(T_2)) \geq s.$$

By dividing by $s - q$ and rounding up the right hand side we get inequality (40). □

Inequalities (40) will be called *design cut-cycle inequalities.*

Theorem 4.13 *Let (W, T_1, T_2) be a cut-cycle configuration such that $\gamma(W) \neq \emptyset$. 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 $r(s-q) \geq \max_{e \in T_2}\{r_e\}$.*
Let $r_W = \begin{cases} 4 \text{ if } D_W \bmod 4 = 0 \\ D_W \bmod 4 \text{ otherwise.} \end{cases}$
Then the inequality

$$x^1(T_1) + r_W x^2(T_1) + r x^1(T_2) + r \times r_W x^2(T_2) \geq \left\lceil \frac{s}{s-q} \times r_W \left\lceil \frac{D_W}{4} \right\rceil \right\rceil \tag{41}$$

is valid for both $MCSIPND_m(G, \mathcal{F}, K)$ and $MCSIPND_s(G, \mathcal{F}, K)$.

Proof. By Theorem 4.10, the following inequalities are valid for both $\text{MCSIPND}_m(G, \mathcal{F}, K)$ and $\text{MCSIPND}_s(G, \mathcal{F}, K)$,

$$\sum_{e \in \delta_{G_{j_i}}(W)} (x_e^1 + r_W x_e^2) \geq r_W \left\lceil \frac{D_W}{4} \right\rceil \qquad \text{for } i = 1, \ldots, s.$$

Also consider the following inequalities which are also valid for both $\text{MCSIPND}_m(G, \mathcal{F}, K)$ and $\text{MCSIPND}_s(G, \mathcal{F}, K)$.

$$(r(s-q) - r_e)x^1(e) \geq 0 \qquad \text{for all } e \in T_2,$$
$$(r(s-q) - r_e)r_W x^2(e) \geq 0 \qquad \text{for all } e \in T_2.$$

By summing these inequalities we obtain

$$\sum_{e\in T_1}\left[(s-q)x_e^1+(s-q)r_W x_e^2\right]+\sum_{e\in T_2}\left[r(s-q)x_e^1+r(s-q)r_W x_e^2\right]\geq s\times r_W\left\lceil\frac{D_W}{4}\right\rceil.$$

By dividing this inequality by $s-q$ and rounding up the right hand side we get inequality (41). □

Inequalities (41) will be called *capacity cut-cycle inequalities*.

We can remark that if $r_W=1$ and $\left\lceil\frac{D_W}{4}\right\rceil=1$ we then obtain the design cut-cycle inequality (40).

4.4 Star-partition inequalities

Let $G=(V,E)$ be a graph and $\mathcal{F}=\{F_1,\ldots,F_t\}$, with $t\geq 2$, 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 $\Lambda=\{e\in E\,|\,e\in[V_k,V_l]\cap F_{j_k}\cap F_{j_l}$, 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 Figure 7).

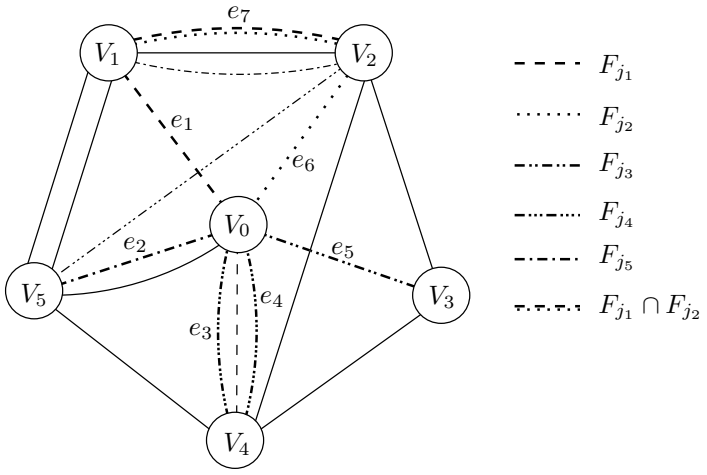


FIG. 7 – A star-partition configuration

Theorem 4.14 *Let $(V_0,V_1,\ldots,V_p,F)$ a star-partition configuration with p odd such that $\gamma(W)\neq\emptyset$. Then inequality*

$$\sum_{l=1,2}x^l(\delta_G(V_0,\ldots,V_p)\backslash F)\geq\left\lceil\frac{p}{2}\right\rceil \tag{42}$$

is valid for both MCSIPND$_m(G,\mathcal{F},K)$ and MCSIPND$_s(G,\mathcal{F},K)$.

Proof. It is clear that the following inequalities are valid for both MCSIPND$_m(G,\mathcal{F},K)$ and MCSIPND$_s(G,\mathcal{F},K)$.

$$\begin{aligned}
\sum_{l=1,2}x^l(\delta_{G\backslash F_{j_i}}(V_i))\geq 1 \qquad & \text{for } i=1,\ldots,p,\\
x^l(e)\geq 0 \qquad & \text{for all } e\in\delta(V_0)\backslash F,\ l=1,2,\\
x^l(e)\geq 0 \qquad & \text{for all } e\in(\delta_G(V_k,V_m)\cap F_{j_k})\backslash F_{j_m},\\
& k=1,\ldots,p,\ m=1,\ldots,p,\ k\neq m,\quad l=1,2.
\end{aligned}$$

By summing these inequalities, we obtain inequality

$$2\sum_{l=1,2} x^l(\delta(V_0,\ldots,V_p)\backslash F) \geq p.$$

By dividing by 2 and rounding up the right hand side , we obtain inequality (42). □

Inequalities (42) will be called *design star-partition inequalities.*

Theorem 4.15 *Let $(V_0, V_1, \ldots, V_p, F)$ be a star-partition configuration.*
Let $r_{V_i} = \begin{cases} 4 \text{ si } D_{V_i} \text{ mod } 4 = 0, \\ D_{V_i} \text{ mod } 4 \text{ sinon,} \end{cases}$ *for* $i = 1, \ldots, p$.
Set

$$r_e = \sum_{\substack{i=1,\ldots,p\ :\\ e\in\delta_{G\backslash F_{j_i}}(V_i)}} r_{V_i}$$

and

$$\lambda_e = \begin{cases} r_e & \text{if } r_e \text{ is even,} \\ r_e+1 & \text{otherwise ,} \end{cases} \quad e \in \delta_G(V_0,\ldots,V_p)\backslash F.$$

Then inequality

$$\sum_{e\in\delta_G(V_0,\ldots,V_p)\backslash F} \left(x_e^1 + \frac{\lambda_e}{2}x_e^2\right) \geq \left\lceil \frac{\sum_{i=1}^p r_{V_i}\left\lceil \frac{D_{V_i}}{4}\right\rceil}{2}\right\rceil \tag{43}$$

is valid for both MCSIPND$_m(G,\mathcal{F},K)$ and MCSIPND$_s(G,\mathcal{F},K)$.

Proof. The following inequalities are valid for both MCSIPND$_m(G,\mathcal{F},K)$ and MCSIPND$_s(G,\mathcal{F},K)$.

$$\sum_{e\in\delta_{G\backslash F_{j_i}}(V_i)} (x_e^1 + r_{V_i}x_e^2) \geq r_{V_i}\left\lceil\frac{D_{V_i}}{4}\right\rceil \quad \text{for } i = 1,\ldots,p,$$

$$\begin{array}{ll}
x_e^1 \geq 0 & \text{for all } e \in \delta_G(V_0,\ldots,V_p)\backslash F, \\
x_e^2 \geq 0 & \text{for } i = 1,\ldots,p, \text{ for all } e \in \delta_G(V_i,V_0)\backslash F, \text{ such that } r_{V_i} \text{ is odd,} \\
x_e^2 \geq 0 & \text{for all } e \in \delta_G(V_k,V_m)\backslash(F_{j_k}\cap F_{j_m}), \text{ such that } r_e \text{ is odd,} \\
 & k = 1,\ldots,p,\ m = 1,\ldots,p,\ k \neq m, \\
x_e^1 \geq 0 & \text{for all } e \in (\delta_G(V_k,V_m)\cap F_{j_k})\backslash F_{j_m}, k = 1,\ldots,p,\ m = 1,\ldots,p,\ k \neq m, \\
x_e^2 \geq 0 & \text{for all } e \in (\delta_G(V_k,V_m)\cap F_{j_k})\backslash F_{j_m}, \text{ such that } r_{V_m} \text{ is odd,} \\
 & k = 1,\ldots,p,\ m = 1,\ldots,p,\ k \neq m.
\end{array}$$

By summing these inequalities we get

$$\sum_{e\in\delta_G(V_0,\ldots,V_p)\backslash F} (2x_e^1 + \lambda_e x_e^2) \geq \sum_{i=1}^s r_{V_i}\left\lceil\frac{D_{V_i}}{4}\right\rceil.$$

As λ_e's, $e \in \delta_G(V_0,\ldots,V_p)\backslash F$ are all even, by dividing this inequality by 2 and rounding up the right hand side we get the inequality

$$\sum_{e\in\delta_G(V_0,\ldots,V_p)\backslash F} \left(x_e^1 + \frac{\lambda_e}{2}x_e^2\right) \geq \left\lceil \frac{\sum_{i=1}^s r_{V_i}\left\lceil \frac{D_{V_i}}{4}\right\rceil}{2}\right\rceil.$$

□

Inequalities (43) are called *capacity star-partition inequalities.*

5 Branch-and-cut and branch-and-cut-and-price algorithms

In this section, we describe four algorithms for the MCSIPND problem. We consider the two variants of the problem (simple and multiple) and for each variant we propose a Branch-and-Cut algorithm based on the node-arc formulation and a Branch-and-Cut-and-Price algorithm based on the path formulation. Our aim is to address the algorithmic applications of the previous results.

We now describe the framework of our algorithms. For the Branch-and-Cut algorithms based on the node-arc formulation, we start the optimization with the linear relaxations of the formulations. The optimal solutions (x^1, x^2, f) of theses relaxations are feasible for the MCSIPND problems if x^1 and x^2 are integral.

For the Branch-and-Cut-and-Price algorithms, we also start the optimization by solving the linear relaxation of the path formulations. For this we use the column generation algorithms described in Section 3. As for the node-arc formulations, the solutions then obtained are feasible for the MCSIPND problems only if x^1 and x^2 are integral.

Usually, a solution (x^1, x^2, f) is not feasible, and thus, in each iteration of the Branch-and-Cut and the Branch-and-Cut-and-Price algorithms, it is necessary to generate further inequalities that are valid for the MCSIPND problem but violated by the current solution (x^1, x^2, f). For this, one has to solve the so-called *separation problem*. This consists, given a class of inequalities, in deciding whether the current solution (x^1, x^2, f) satisfies the inequalities, and if not, in finding an inequality that is violated by (x^1, x^2, f). An algorithm which solves this problem is called a *separation algorithm*. The inequalities given above are all valid for the four polytopes MCSIPND_m^{na}, MCSIPND_m^{p}, MCSIPND_s^{na} et MCSIPND_s^{p} except the saturation inequalities which are valid only for the simple version of the problem. Hence theses inequalities are used in our algorithms. The separation is performed in the following order :

1. arc residual capacity constraints (9) (for the Branch-and-Cut algorithms only),
2. design cut constraints (32) and capacity demand cut constraints (33)-(37),
3. saturation constraints (39) (for MCSIPND_s only),
4. design cut-cycle constraints (40),
5. capacity cut-cycle constraints (41),
6. design star-partition constraints (42),
7. capacity star-partition constraints (43).

We remark that all inequalities are global (*i.e.* valid in all the Branch-and-Cut tree and the Branch-and-Cut-and-Price tree) and several constraints may be added at each iteration. Moreover, we go to the next class of inequalities only if we have found no violated inequality. Our strategy is to try to detect violated constraints at each node of the 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 at most 200 inequalities at a time.

Now we describe the separation procedures used in our algorithms. All our separation algorithms are applied on $G_{(\bar{x}^1,\bar{x}^2)} = (V, E_{(\bar{x}^1,\bar{x}^2)})$ where $(\bar{x}^1, \bar{x}^2)$ is the restriction on x^1 and x^2 of the current LP solution, and $E_{(\bar{x}^1,\bar{x}^2)}$ contains all the edges uv of E such that $\bar{x}^1_{uv} + \bar{x}^2_{uv} \neq 0$.

To separate the design cut inequalities and the capacity demand cut inequalities, we have developed a fast heuristic. We first check whether a degree cut $\delta_{G_{(\bar{x}^1,\bar{x}^2)}}(v)$, $v \in V$, is violated. Then we start contracting edges uv with high value $2.5\bar{x}^1_{uv} + 10\bar{x}^2_{uv} - \sum_{k\in\gamma(\{u\})\cup\gamma(\{v\})} \omega_k$ until we get a graph on two nodes. In each iteration we check if the cut associated with the node arising from the contraction, induces a violated constraint of type (33), (35), (36) or (37).

When this heuristic does not any more allow to find violated inequalities, we compute the so-called Gomory-Hu tree [17] on the graph $G_{(\bar{x}^1,\bar{x}^2)}$ with the weight $\bar{x}^1_{uv} + \bar{x}^2_{uv}$ for each edge uv. This

tree has 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 [21] which requires $|V|-1$ maximum flow computations. The maximum flow computations are handled by the efficient Goldberg and Tarjan algorithm [16] that runs in $O(mnlog\frac{n^2}{m})$ time where m and n are the number of edges and nodes of G, respectively. Then we calculate the right hand side for all the cuts in the Gomory-Hu tree and check if the found constraints are violated.

During the separation of the saturation constraints (39), we consider the cuts $\delta_{G_{(\bar{x}^1,\bar{x}^2)}}(v)$, $v \in V$. We test if these inequalities are violated and if so we add them to the program. We don't consider the other cuts of the graph $G_{(\bar{x}^1,\bar{x}^2)}$ induced by more than two nodes which seem almost never be violated.

Now we turn our attention to the separation of the cut-cycle inequalities (40) and (41). For more efficiency, we have used these constraints only when $q = 1$. In fact we remarked that the design cut-cycle inequalities and the capacity cut-cycle inequalities which are violated are usually of this type.

To separate the design cut-cycle constraints with $q = 1$, we compute the Gomory-Hu tree of the graph $G_{(\bar{x}^1,\bar{x}^2)}$ with the weight for each edge uv equal to the sum $\bar{x}^1_{uv} + \bar{x}^2_{uv}$. Then for each cut given by the Gomory-Hu tree, with value less than 2, we test if the cut intersects at least one demand. If this is the case, then it yields a design cut-cycle inequality (40) violated by $(\bar{x}^1, \bar{x}^2)$. Then sets T_1 and T_2 are determined so that T_1 is maximal, using the following greedy procedure (Algorithm 1). Since the Gomory-Hu algorithm runs with a large complexity, in order to accelerate our separation for the cut-cycle inequalities, we first consider the degree cuts $\delta_{G_{(\bar{x}^1,\bar{x}^2)}}(v)$, $v \in V$. The computation of the Gomory-Hu tree is considered only if no cuts of this type of value less than 2 are found.

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

For the separation of the capacity cut-cycle constraints, we first consider the degree cuts $\delta_{G_{(\bar{x}^1,\bar{x}^2)}}(v)$, $v \in V$. We calculate the right hand side. If the associated constraint is violated, we then determine the sets T_1 et T_2 using Algorithm 1. Then we start contracting edges until we get a graph on two nodes. In each iteration we contract an edge uv with the biggest value for $2.5\bar{x}^1_{uv} + 10\bar{x}^2_{uv} - \sum_{k \in \gamma(\{u\}) \cup \gamma(\{v\})} \omega_k$ and check whether the new node obtained by contraction together with T_1 and T_2 induces a violated capacity cut-cycle inequality.

We now discuss our separation routine for the star-partition inequalities (42) and (43). Our algorithm consists in determining fractional cycles in the supporting graph, satisfying some conditions. These cycles have to be odd, in order to have a chance to find a violated design star-partition inequality. 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 a way that either the design star-partition inequality or the capacity star-partition 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 $(\bar{x}^1, \bar{x}^2)$.

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.

In the following section, we give some computational results obtained with the algorithms presented above for random instances and for real instances provided by France Télécom.

6 Computational results

The Branch-and-Cut and Branch-and-Cut-and-Price algorithms described in the previous section have been implemented in C++, using ABACUS (A Branch-And-CUt System) 2.4 alpha [1, 14, 42] to manage the Branch-and-Cut tree and Cplex 9.0 as LP-solver [11]. 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 ([38]) by randomly generating the node set, the edge sets F_e and the set of demands K. For all the instances, the graph G^1, representing the IP network, is considered complete.

These instances were generated with 6, 8 and 10 nodes, $|\mathcal{F}| = 10, 20$ and $|K| = 5, 10, 20$. Five instances of each size, each $|\mathcal{F}|$ and each $|K|$ 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 6 to 18 nodes and $\mathcal{F}$ with 11 to 32 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 router 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.

The number of commodities is between 5 and 20. We randomly generated the extremities of the commodities. The amount of each commodity is calculated with the gravity model which uses the distance and the population of both the origin and destination cities. The general expression of the gravitaty model for a commodity (o_k, d_k, v_k) of K is $v_k = \frac{P_{o_k}^{\alpha} P_{d_k}^{\beta}}{d_{o_k,d_k}}$ where P_{o_k} and P_{d_k} are the populations of the origin and destination cities, respectively and d_{o_k,d_k} represents the euclidian distance between o_k and d_k. We have fixed $\alpha = 1.2$ and $\beta = 0.8$ to have different volume between two towns in the two directions.

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

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

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

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 $\kappa(f)$ 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 $\kappa(f)$ proportional to the number of the optical segments on P_f. So, a first natural function $\kappa(f)$ 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 $c + |P_f|$.

In the various tables, the entries are :

$\|V^1\|$	:	the number of nodes of G^1,
$\|\mathcal{F}\|$	:	the number of sets F_e,
$\|K\|$	:	the number of demands,
Algo	:	the type of algorithm (na (resp. p means that the used algorithm is based on the
	:	node-arcs (resp. path) formulation.
FV	:	the number of flow variables,
NC	:	the number of generated cut inequalities,
NRC	:	the number of generated arc residual capacity inequalities,
NS	:	the number of generated saturation inequalities (only for MCSIPND$_s$ problem),
NCC	:	the number of generated cut-cycle inequalities,
NSP	:	the number of generated star-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,
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 concerns the problem MCSIPND$_m$ (with multiple links) for the random instances. In these experiments, we have considered three instances for each size. Table 1 reports the average results obtained for these instances with both algorithms based on the node-arc and the path formulations.

As we can observe, all the instances with 6 nodes have been solved to optimality. Moreover, they have been solved in less than 5 minutes (and in less than 2 minutes using the path formulation based algorithm), except those with $|\mathcal{F}| = 20$ and $|K| = 20$ which needed around 33 minutes to be solved to optimality using the node-arc formulation based algithm. We also remark that for the instances with a reduced number of commodities, the problem seems to be much easier to solve. In fact, for all the instances with 5 commodities, less than one hour was needed to get the optimal solution except for the instances with 10 nodes and $|\mathcal{F}| = 20$.

The instances with 10 and 20 commodities and $|\mathcal{F}| = 20$ seem to be harder to solve. In fact, none of these instances could be solved to optimality with the node-arc formulation within the time limit, when the number of nodes exceeds 8. The results are more promising with the path formulation where 5 among the 12 instances have nevertheless been solved in an average time sometimes less than two hours.

Among the 54 tested instances, 17 could not be solved within the time limit of five hours using the node-arc formulation based algorithm, and only 12 instances with that based on the path formulation. So, this would imply that the latter algorithm is more efficient. However for 10 nodes, $|\mathcal{F}| = 10$ and $|K| = 20$, two instances have been solved with the first algorithm but they did not with the second one.

The number of flow variables of the node-arc formulation is fixed and depends only on $|V^1|$, $|\mathcal{F}|$ and $|K|$. This is not however the case for the flow variables of the path formulation. We remark

$\|V^1\|$	$\|\mathcal{F}\|$	$\|K\|$	Algo	FV	NC	NRC	NCC	NEP	NT	o/p	Gap	TT
6	10	5		1500	85.67	78.00	0.67	0.67	132.33	3/3	9.47	0 :00 :13.52
6	10	10	*na*	3000	80.00	89.00	0.00	0.00	109.67	3/3	5.86	0 :00 :40.11
6	10	20		6000	71.33	201.00	0.33	0.00	123.00	3/3	6.39	0 :02 :22.54
6	10	5		590.67	93.33	-	0.33	0.67	112.33	3/3	9.40	0 :00 :07.25
6	10	10	*p*	899.33	92.67	-	0.00	0.00	95.00	3/3	5.24	0 :00 :08.38
6	10	20		1363.33	124.67	-	0.00	0.00	157.00	3/3	5.96	0 :00 :15.40
6	20	5		3000	105.00	226.33	2.00	0.67	220.33	3/3	10.74	0 :01 :38.04
6	20	10	*na*	6000	125.67	427.67	2.67	0.00	214.33	3/3	8.41	0 :04 :27.97
6	20	20		12000	153.33	641.00	0.33	0.00	347.00	3/3	7.46	0 :33 :29.84
6	20	5		813.33	118.33	-	1.67	0.00	237.67	3/3	10.71	0 :00 :33.72
6	20	10	*p*	1490.00	156.67	-	1.00	0.00	165.67	3/3	8.29	0 :00 :39.12
6	20	20		2326.00	156.33	-	0.33	0.00	297.00	3/3	7.38	0 :01 :45.86
8	10	5		2800	104.00	84.00	0.67	0.33	115.00	3/3	12.09	0 :00 :38.38
8	10	10	*na*	5600	304.33	449.67	2.00	1.00	771.00	3/3	10.68	0 :16 :57.89
8	10	20		11200	381.33	1290.00	0.00	0.00	2585.00	1/3	9.60	4 :13 :51.50
8	10	5		1094.00	91.67	-	0.33	0.33	121.67	3/3	11.35	0 :00 :22.58
8	10	10	*p*	2769.33	457.00	-	0.67	0.00	876.33	3/3	10.31	0 :04 :58.44
8	10	20		4100.00	2083.00	-	1.33	0.00	3334.33	3/3	9.48	0 :22 :59.97
8	20	5		5600	282.67	1102.33	2.00	0.67	1041.00	3/3	14.84	0 :30 :23.18
8	20	10	*na*	11200	281.00	1524.67	2.00	0.67	2454.00	0/3	25.13	5 :00 :00.00
8	20	20		22400	283.00	3726.00	1.00	0.00	1266.33	0/3	15.30	5 :00 :00.00
8	20	5		3324.67	246.33	-	2.67	1.00	1223.67	3/3	11.89	0 :21 :40.39
8	20	10	*p*	5937.33	5472.00	-	1.67	2.00	9515.67	2/3	13.63	3 :11 :59.93
8	20	20		5619.00	1733.33	-	0.67	0.00	5004.67	2/3	10.64	1 :57 :35.10
10	10	5		4500	819.67	519.33	0.33	0.33	2282.33	3/3	13.58	0 :34 :04.75
10	10	10	*na*	9000	846.33	1188.67	1.00	0.00	1773.67	3/3	9.81	1 :39 :03.70
10	10	20		18000	524.67	1325.33	0.33	0.67	1144.00	2/3	33.42	3 :22 :05.67
10	10	5		7192.67	1271.33	-	0.33	0.00	2159.00	3/3	11.49	0 :52 :43.15
10	10	10	*p*	8197.33	2775.00	-	0.67	0.00	3378.33	3/3	9.98	1 :45 :52.21
10	10	20		12500.00	2773.67	-	1.00	0.00	11246.67	0/3	12.38	5 :00 :00.00
10	20	5		9000	199.33	2140.00	0.67	0.00	2653.00	1/3	38.60	4 :05 :09.57
10	20	10	*na*	18000	378.33	5020.00	0.67	0.00	1523.67	0/3	33.59	5 :00 :00.00
10	20	20		36000	491.00	4169.33	0.67	0.00	457.00	0/3	45.39	5 :00 :00.00
10	20	5		6991.00	1249.33	-	3.67	5.00	3978.33	1/3	33.94	3 :59 :39.57
10	20	10	*p*	11784.00	5659.67	-	2.67	2.33	3953.00	1/3	26.08	4 :33 :09.10
10	20	20		14427.33	3986.67	-	1.00	0.67	3665.00	0/3	33.98	5 :00 :00.00

TAB. 1 – Result for random instances for the MCSIPND$_m$ problem

that the number of the flow variables generated, using this formulation, is generally smaller than that of the node-arc formulation. This may explain the fact that some instances could be solved to optimality using the path formulation whereas they could not using the node-arc one. Moreover the CPU times obtained with the path formulation based algorithm are less than those obtained with the node-arc formulation based algorithm except for the instances with 10 nodes and $|\mathcal{F}| = 10$. The difference of time is sometimes big. For example, for the instances with 6 nodes, $|\mathcal{F}| = 20$ and $|K| = 20$, the time goes from less than 2 minutes to more than 30 minutes and for 8 nodes, $|\mathcal{F}| = 10$ and $|K| = 20$ it goes from 20 minutes to more than 4 hours. Therefore the path formulation seems to give better results than the node-arc formulation.

We can also remark that for most of the instances, a significant number of cut and arc residual capacity inequalities (for the node-arc formulation) have been generated. This implies that these inequalities are useful for the random problems. However the cut-cycle and 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 cut-cycle and star-partition configurations.

Table 2 presents the results for the MCSIPND$_m$ problem for the real instances with both path and node-arc formulation based algorithms. Here the cost function represents, for each link of the IP network, the number of links of the optical network used in the associated path. This cost function is then integral.

It appears from the table that the difficulty to solve the instances increases with the number of

$\|V^1\|$	$\|\mathcal{F}\|$	$\|K\|$	Algo	FV	NC	NRC	NCC	NEP	NT	Copt	Gap	TT
6	11	5		1650	10	10	0	1	3	57	0.00	0 :00 :00.77
6	11	10	*na*	3300	56	111	2	3	65	99	8.08	0 :00 :27.28
6	11	15		4950	99	779	1	0	157	105	7.62	0 :02 :41.68
6	11	20		6600	199	1194	1	0	375	115	9.57	0 :13 :34.46
6	11	5		137	15	-	0	0	3	57	0.00	0 :00 :00.32
6	11	10	*p*	628	59	-	2	1	81	99	8.08	0 :00 :07.12
6	11	15		1071	88	-	1	0	147	105	7.62	0 :00 :17.86
6	11	20		1778	233	-	1	0	451	115	9.57	0 :01 :37.51
8	17	5		4760	173	114	2	0	115	81	11.11	0 :02 :36.71
8	17	10	*na*	9520	191	717	2	7	567	109	15.60	0 :49 :59.51
8	17	15		14280	512	815	4	2	621	123	13.01	2 :12 :27.03
8	17	20		19040	666	9743	7	5	709	*160*	33.75	5 :00 :00.00
8	17	5		1693	178	-	1	0	93	81	11.11	0 :00 :55.39
8	17	10	*p*	3452	180	-	2	5	449	109	15.60	0 :07 :13.23
8	17	15		4717	511	-	4	2	1155	123	14.63	0 :18 :48.43
8	17	20		8230	2570	-	5	19	5655	131	19.08	3 :34 :26.29
10	25	5		11250	71	157	4	0	113	129	7.75	0 :16 :27.52
10	25	10	*na*	23400	262	279	4	2	369	156	3.21	2 :45 :10.77
10	25	15		33750	772	1616	6	0	473	*231*	18.18	5 :00 :00.00
10	25	20		45000	447	2754	6	1	247	*297*	34.34	5 :00 :00.00
10	25	5		4011	99	-	2	0	151	129	8.53	0 :08 :16.44
10	25	10	*p*	5594	318	-	2	4	373	156	3.21	0 :43 :35.76
10	25	15		8673	5497	-	5	0	3993	*213*	11.27	5 :00 :00.00
10	25	20		8966	4038	-	9	3	3317	*261*	24.90	5 :00 :00.00
12	32	5		21120	85	180	5	0	99	129	8.53	0 :54 :33.54
12	32	10	*na*	42240	333	777	5	1	303	*194*	12.37	5 :00 :00.00
12	32	15		63360	251	1129	8	0	127	*300*	32.67	5 :00 :00.00
12	32	20		84480	357	2001	7	0	89	*392*	41.58	5 :00 :00.00
12	32	5		7188	100	-	2	0	153	129	8.53	0 :34 :49.36
12	32	10	*p*	15120	572	-	4	1	559	*182*	6.59	5 :00 :00.00
12	32	15		8852	262	-	5	2	127	*285*	28.77	5 :00 :00.00
12	32	20		13988	1109	-	7	0	681	*378*	39.15	5 :00 :00.00
14	27	5	*na*	24570	55	22	0	0	73	150	8.00	0 :34 :17.06
14	27	10		49140	342	298	2	0	175	198	6.06	4 :56 :12.04
14	27	5	*p*	4081	44	-	0	0	49	150	8.00	0 :10 :56.41
14	27	10		17610	608	-	1	0	205	*198*	6.06	5 :00 :00.00
16	29	5	*na*	34800	315	208	3	0	323	*153*	11.76	5 :00 :00.00
16	29	10		69600	88	206	1	0	67	185	4.32	4 :21 :11.28
16	29	5	*p*	15910	294	-	3	0	177	*159*	15.09	5 :00 :00.00
16	29	10		19934	114	-	2	0	77	185	4.32	3 :26 :08.28
18	30	5	*na*	45900	87	234	2	0	229	*218*	17.43	5 :00 :00.00
18	30	5	*p*	21494	47	-	0	0	63	*220*	18.64	5 :00 :00.00

TAB. 2 – Results for real instances for the MCSIPND$_m$ problem

nodes and the number of commodities in the network. All the instances with 5 commodities and up to 14 nodes have been solved in less than one hour. Also for the instances with 10 commodities, we have obtained an optimal solution for all the instances with no more than 10 nodes in less than 3 hours with the node-arc formulation and in less than 45 minutes with the path formulation. The instance with 16 nodes and 10 commodities has also been solved to optimality.

In addition, we remark that for the instances with 15 and 20 commodities only those with no more than 8 nodes could be solved to optimality. For all the instances which have not been solved in the time limit, we have nevertheless obtained feasible solutions (given in italic). Also, we can remark that for some instances, we have obtained a relatively small gap. This is for example the case for the instance with 12 nodes and 10 commodities where the gap is 6%. For the larger instances, the gap does not exceed 18%.

Moreover we notice that all the instances solved with the algorithm based on the node-arc formulation have also been solved with the one based on the path formulation, except that with 14 nodes and 10 commodities. Furthermore, the time needed was much lower. In fact, as it can be observed, for the instance with 8 nodes and 15 commodities, the time passes from 2 hours 12 minutes to 18 minutes and for that with 10 nodes and 10 commodities, it passes from 2 hours 45

minutes to 43 minutes. As regards the instances which have not been solved with the two algorithms in the time limit, we can remark that the value of the best feasible solution we have found with the path formulation is cheaper than that obtained with the node-arc one and this is for most of the instances.

As for the random problems, for most of the real instances, a significant number of cut inequalities have been generated. It also appears that the cut-cycle inequalities and star-partition inequalities are more effective for this type of instances. This can be explained by the fact that the way the sets F_e's are built, it is easier to find cut-cycle and star-partition configurations. It seems that the node-arc formulation permits to generate more cut-cycle inequalities (72 constraints generated for all the instances against 58 with the path formulation) and the path formulation privileges rather the star-partition inequalities (37 constraints generated against only 22 with the node-arc formulation).

For these instances, we can also observe that the number of flow variables generated for the path formulation is much lower than the one used in the node-arc one. For example, the instance with 16 nodes and 10 commodities has used 69600 flow variables for the first formulation and only 19934 flow variables in the second one.

$\|V^1\|$	$\|\mathcal{F}\|$	$\|K\|$	Algo	FV	NC	NRC	NS	NCC	NEP	NT	o/p	Gap	TT
6	10	5		1500	58.67	49.67	3.67	0.33	0.00	46.33	3/3	14.04	0 :00 :07.70
6	10	10	*na*	3000	43.67	47.33	4.67	0.00	0.00	23.67	3/3	4.48	0 :00 :15.00
6	10	20		6000	48.67	96.67	3.33	0.00	0.00	36.33	3/3	6.63	0 :01 :17.19
6	10	5		550.67	85.00	-	3.33	0.33	0.33	52.33	3/3	13.93	0 :00 :04.58
6	10	10	*p*	776.67	39.57	-	4.67	0.33	0.00	29.67	3/3	4.54	0 :00 :02.95
6	10	20		836.67	21.00	-	1.33	0.00	0.00	17.00	3/3	4.93	0 :00 :02.39
6	20	5		3000	75.67	151.00	3.00	1.67	0.33	106.33	3/3	16.16	0 :01 :04.65
6	20	10	*na*	6000	70.67	187.33	9.67	1.00	0.00	41.00	3/3	9.93	0 :01 :29.15
6	20	20		12000	44.67	94.33	2.67	0.00	0.00	24.33	3/3	6.22	0 :03 :11.87
6	20	5		812.00	55.67	-	3.00	2.00	0.00	93.67	3/3	15.82	0 :00 :16.27
6	20	10	*p*	1194.33	49.33	-	8.67	2.00	0.00	50.33	3/3	9.02	0 :00 :10.58
6	20	20		1532.33	44.33	-	3.67	0.00	0.00	24.33	3/3	7.58	0 :00 :08.16
8	10	5		2800	81.67	97.33	0.67	0.33	0.33	85.00	3/3	13.16	0 :00 :45.93
8	10	10	*na*	5600	215.33	293.00	2.33	0.67	0.00	296.33	3/3	10.02	0 :06 :34.95
8	10	20		11200	277.00	773.33	4.33	0.00	0.00	373.67	3/3	9.02	0 :30 :53.21
8	10	5		1225.33	90.67	-	1.00	1.00	0.67	81.67	3/3	12.42	0 :00 :20.48
8	10	10	*p*	1910.67	259.00	-	2.33	2.33	0.00	270.33	3/3	9.16	0 :01 :15.36
8	10	20		2912.00	328.67	-	4.33	0.00	0.00	309.67	3/3	8.76	0 :02 :02.43
8	20	5		5600	177.33	570.33	1.00	1.67	0.67	118.33	3/3	13.43	0 :05 :30.62
8	20	10	*na*	11200	381.00	1140.00	11.00	0.67	0.67	772.33	3/3	14.34	1 :38 :42.90
8	20	20		22400	305.67	1988.00	13.00	0.33	0.00	639.67	1/3	12.44	3 :50 :48.94
8	20	5		2514.67	164.67	-	1.33	3.00	0.33	135.67	3/3	12.71	0 :02 :30.74
8	20	10	*p*	4944.00	623.67	-	10.00	1.00	0.33	1890.33	3/3	14.12	0 :53 :31.81
8	20	20		5373.00	425.00	-	17.33	0.33	0.00	1000.33	3/3	10.79	0 :32 :17.99
10	10	5		4500	189.67	510.00	0.00	0.00	0.00	412.00	3/3	18.06	0 :06 :52.79
10	10	10	*na*	9000	171.33	610.00	2.00	0.00	0.00	1379.00	3/3	11.49	1 :39 :00.75
10	10	20		18000	176.67	823.33	0.00	1.00	0.33	1214.00	2/3	15.42	2 :54 :21.12
10	10	5		7362.00	1036.33	-	0.00	0.33	0.00	2625.00	2/3	15.75	1 :46 :59.66
10	10	10	*p*	8874.00	873.00	-	0.33	2.00	0.00	2756.33	3/3	12.60	2 :02 :09.73
10	10	20		12624.00	1534.33	-	0.67	1.33	0.00	7562.67	1/3	11.87	2 :41 :08.19
10	20	5		9000	314.00	1293.00	0.00	3.00	0.00	507.00	1/3	22.14	3 :29 :34.19
10	20	10	*na*	18000	740.00	5198.00	0.00	1.00	0.00	1463.00	0/3	34.28	5 :00 :00.00
10	20	20		36000	448.67	4156.33	1.67	0.67	0.00	464.33	0/3	27.72	5 :00 :00.00
10	20	5		10438.67	722.67	-	1.33	6.67	1.33	2799.67	1/3	22.90	3 :29 :28.52
10	20	10	*p*	13042.00	2816.67	-	0.00	4.33	2.00	*3395.67*	0/3	15.92	5 :00 :00.00
10	20	20		14828.00	3292.33	-	3.33	1.33	0.00	*3321.00*	0/3	17.82	5 :00 :00.00

TAB. 3 – Result for random instances for the MCSIPND$_s$ problem

Tables 3 and 4 report the results for the MCSIPND$_s$ problem (single version). As in the previous experiments related to the MCSIPND$_m$ problem, we have considered random and real instances. We have tested our two algorithms on the same instances. Table 3 reports the results for the random instances. As for the multiple case, we have tested three instances for each size, Table 3 gives the average result. Also here we have used the saturation constraints (39) which are valid for this variant of the problem.

We can note that all the instances with 6 and 8 nodes have been solved to optimality with at least one of the algorithms. This is also the case for all the instances with $|\mathcal{F}| = 10$ except that with 10 nodes and 20 commodities, and for the instances with $|K| = 5$, except two of the instances with 10 nodes and $|\mathcal{F}| = 20$.

Among the 9 instances with 10 nodes and $|\mathcal{F}| = 20$, only one has been solved in the time limit of 5 hours. The gap obtained is generally not very hight (less than 18%). Also here both algorithms seem to have similar performance. In fact most of the instances solved by one have also been solved by the other.

$\|V^1\|$	$\|\mathcal{F}\|$	$\|K\|$	Algo	FV	NC	NRC	NS	NCC	NEP	NT	Copt	Gap	TT
6	11	5		1650	12	21	0	0	1	3	57	0.00	0 :00 :06.34
6	11	10	*na*	3300	46	137	1	2	0	79	104	10.58	0 :00 :58.33
6	11	15		4950	71	389	1	1	0	75	109	11.01	0 :01 :39.03
6	11	20		6600	72	705	1	1	0	191	122	9.84	0 :09 :39.32
6	11	5		103	4	-	2	4	0	1	57	0.00	0 :00 :00.24
6	11	10	*p*	618	31	-	1	3	0	89	104	10.58	0 :00 :06.05
6	11	15		931	48	-	1	2	0	57	109	10.09	0 :00 :07.15
6	11	20		1496	47	-	1	3	0	173	122	9.84	0 :00 :27.73
8	17	5		4760	187	101	1	1	0	97	82	12.20	0 :01 :45.53
8	17	10	*na*	9520	165	955	4	5	4	777	115	20.00	1 :23 :37.11
8	17	15		14280	320	3983	3	8	0	981	129	22.48	4 :00 :51.52
8	17	20		19040	550	8849	3	10	3	771	*143*	25.87	5 :00 :00.00
8	17	5		1549	109	-	1	2	0	61	82	12.20	0 :00 :27.62
8	17	10	*p*	2950	118	-	2	6	6	1069	115	20.00	0 :12 :13.75
8	17	15		4267	226	-	3	12	1	777	129	19.38	0 :17 :21.46
8	17	20		5856	391	-	3	12	7	731	135	20.74	0 :25 :56.44
10	25	5		11250	44	135	2	4	0	31	129	8.53	0 :04 :04.37
10	25	10	*na*	23400	82	47	1	0	0	17	157	3.82	0 :05 :29.97
10	25	15		33750	553	1743	2	4	0	485	*223*	13.90	5 :00 :00.00
10	25	20		4500	336	3346	2	6	0	227	*261*	24.52	5 :00 :00.00
10	25	5		3043	37	-	2	5	0	73	129	6.98	0 :02 :59.43
10	25	10	*p*	1316	69	-	1	1	0	15	157	3.82	0 :00 :08.57
10	25	15		8220	1484	-	2	8	0	2015	207	7.25	2 :42 :47.08
10	25	20		10380	1105	-	2	8	0	1457	*243*	18.93	5 :00 :00.00
12	32	5		21120	49	215	1	4	0	37	129	7.75	0 :21 :12.29
12	32	10	*na*	42240	233	485	1	4	0	291	*186*	8.60	5 :00 :00.00
12	32	15		63360	211	1125	1	4	0	139	*247*	18.22	5 :00 :00.00
12	32	20		84480	224	1087	1	7	0	79	*317*	27.44	5 :00 :00.00
12	32	5		6497	84	-	1	4	0	97	129	8.53	0 :17 :26.27
12	32	10	*p*	13723	421	-	1	4	0	851	*182*	6.59	5 :00 :00.00
12	32	15		15291	512	-	1	10	6	813	*224*	9.82	5 :00 :00.00
12	32	20		12114	615	-	1	12	0	823	*292*	20.55	5 :00 :00.00
14	27	5	*na*	24570	57	33	0	0	0	41	150	4.67	0 :19 :08.56
14	27	10		49140	512	387	0	0	0	231	*226*	17.26	5 :00 :00.00
14	27	5	*p*	4326	53	-	0	0	0	27	150	5.33	0 :07 :44.68
14	27	10		18212	652	-	1	4	0	281	*208*	10.10	5 :00 :00.00
16	29	5	*na*	34800	343	316	0	1	0	265	*158*	13.92	5 :00 :00.00
16	29	10		69600	85	171	0	1	0	45	191	5.76	2 :16 :08.88
16	29	5	*p*	18949	338	-	0	3	0	119	*158*	13.92	5 :00 :00.00
16	29	10		13563	80	-	0	3	0	23	191	5.76	0 :17 :14.14
18	30	5	*na*	45900	68	262	1	3	0	229	*211*	14.69	5 :00 :00.00
18	30	5	*p*	23332	43	-	1	3	0	77	*231*	22.08	5 :00 :00.00

TAB. 4 – Results for real instances for the MCSIPND$_s$ problem

The saturation constraints appear in a small proportion with respect to the cut and arc residual capacity inequalities. This does not imply that these inequalities are not necessary for this variant of the problem. On the contrary, they have been quite utile for solving the random instances. Also we notice that for the multiple variant of the problem, the number of generated cut-cycle and star-partition inequalities is also not very significant in this case.

Table 4 presents the results obtained for the MCSIPND$_s$ problem for the real instances. We can remark that in general, the results based on the path formulation, obtained with the Branch-and-Cut-and-Price algorithm, are better than those based on the node-arc formulation. In fact, several

instances which have not been solved in the time limit with the latter formulation, have been solved with the former one, and even in less than 30 minutes for some of them like that with 8 nodes and 20 commodities. Furthermore, for the path formulation based algorithm, the CPU time is much smaller. Also for most of the instances which could not be solved in the time limit, the gap has significantly descreased using the path formulation, as for example for the instance with 12 nodes and 15 commodities, the gap passes from 18,22 to 9,82.

It seems however that the MCSIPND problem in its simple version is easier to solve. In fact, for the random instances and the node-arc formulation, only 11 among the 54 instances could not be solved for the MCSIPND$_s$ problem against 17 for the MCSIPND$_m$ problem. The difference is less important for the path formulation. Here the number of solved instances is almost the same, but the CPU time needed for the MCSIPND$_s$ problem is generally lower.

We can also remark that the optimal solution for the MCSIPND$_m$ problem has a cost lower than that of the MCSIPND$_s$. This is because each feasible solution for the latter problem is also feasible for the former one.

For several instances presented in the previous tables, we have not obtained an optimal solution. In order to evaluate the performance of our algorithms in providing provably near-optimal solutions in reasonnable time, we have noted the value of the best feasible solution after each hour of computation. Table 5 shows the evolution of the current feasible solution for these instances. The differents columns of this table represent :

$\|V^1\|$	:	the number of nodes of G^1,
$\|\mathcal{F}\|$	:	the number of sets F_e,
$\|K\|$	:	the number of demands,
Algo	:	the type of algorithm (na (resp. p) means thatthe used algorithm is based on the
	:	node-arc (resp. path) formulation.
Pb	:	the problem (m represents the multiple version and s the simple one)
BS	:	the value of the best founded solution,
Gap2	:	the relative error between the best feasible solution and the best lower bound.

					After 1h		After 2h		After 3h		After 4h		After 5h	
$\|V^1\|$	$\|\mathcal{F}\|$	$\|K\|$	Algo	Pb	BS	Gap2	BS	Gap2	BS	Gap2	BS	Gap2	BS	Gap2
10	25	15	na	m	243	25,91	231	19,07	231	18,46	231	18,46	231	18,46
10	25	15	p	m	218	10,66	218	10,10	216	8,54	213	7,04	213	7,04
12	32	10	na	m	272	57,23	207	18,29	207	17,61	194	10,23	194	9,60
12	32	10	p	m	189	8,00	182	2,82	182	2,82	182	2,25	182	2,25
14	27	10	p	m	246	26,15	198	1,02	198	1,02	198	1,02	198	0,51
16	29	5	na	m	202	36,49	159	7,43	159	6,00	153	2,00	153	1,32
16	29	5	p	m	202	36,49	167	12,84	167	12,84	167	12,08	159	6,71
18	30	5	na	m	323	73,66	264	39,68	239	25,13	218	13,54	218	12,95
18	30	5	p	m	222	21,98	222	20,65	222	19,35	222	17,46	222	17,46
10	25	15	na	s	240	21,21	227	13,50	227	12,94	227	12,94	223	10,95
12	32	10	na	s	217	24,71	201	14,20	193	8,43	186	3,91	186	3,91
12	32	10	p	s	188	5,62	182	1,68	182	1,11	182	1,11	182	1,11
14	27	10	na	s	226	17,10	226	15,90	226	13,57	226	13,00	226	12,44
14	27	10	p	s	225	12,50	225	11,94	214	5,94	208	2,46	208	1,96
16	29	5	na	s	163	7,95	163	5,84	163	5,16	158	1,28	158	0,64
16	29	5	p	s	169	14,19	169	11,92	163	7,24	163	6,54	163	5,84
18	30	5	na	s	241	25,52	211	8,21	211	7,11	211	6,57	211	6,03
18	30	5	p	s	231	20,94	231	20,31	231	19,69	231	19,69	231	19,07

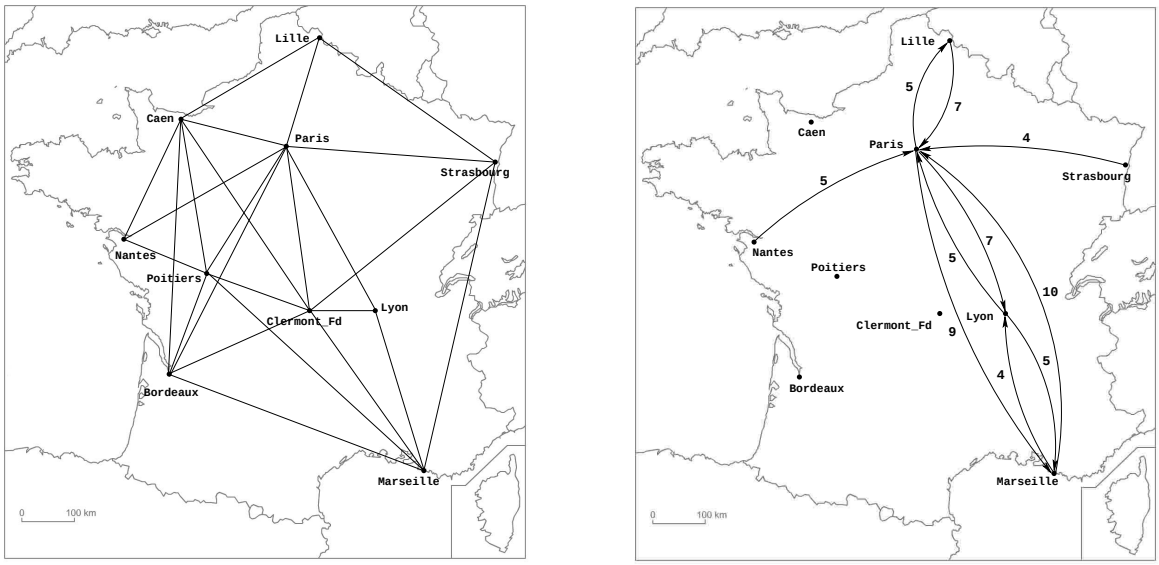
TAB. 5 – Unsolved instances

The first part of the table concerns the MCSIPND$_m$ problem whereas the second one is related to the MCSIPND$_s$ one. The lines are presented in pairs, one for the node-arc formulation and one for the path formulation. When there is only one line, this means that one of the formulation has permited to find the optimal solution in the time limit.

We remark that for most of the instances, we have a very near-optimal solution. For example, for the MCSIPND$_s$ problem and the instance with 16 nodes, the gap is 1,32%. However for harder instances, the best feasible solution is still distant of the optimal one. For example, the gap for the instance with 18 nodes is near 15% for the MCSIPND$_m$ problem and 6% (resp. 19%) for the MCSIPND$_s$ problem for the node-arc formulation (resp. the path formulation).

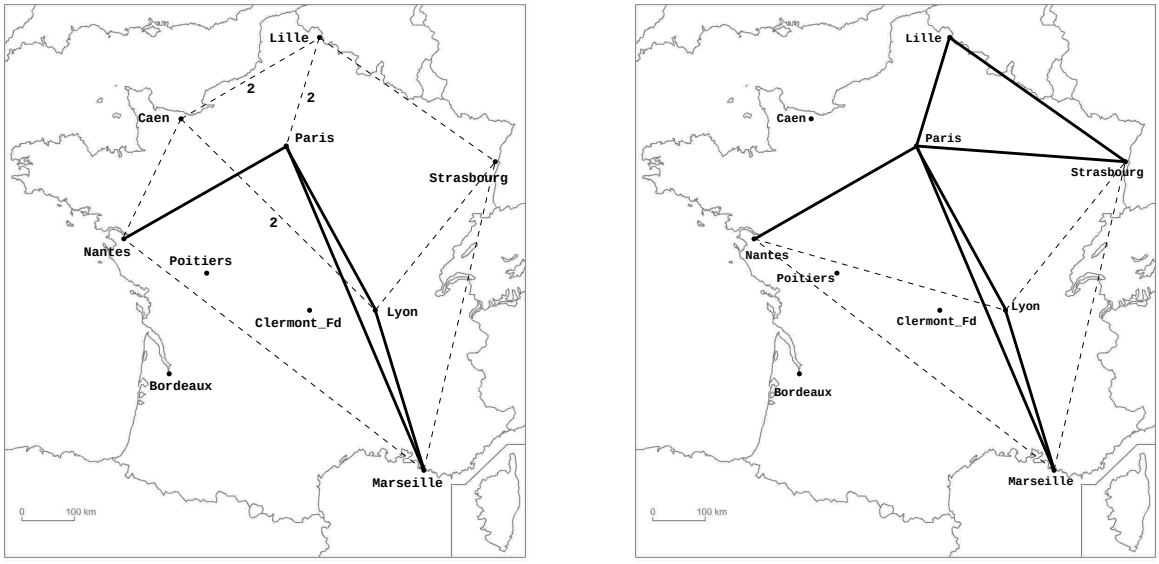
We can note that, as time elapsed, the gap decreases. Actually we have observed that the best solution decreases and the global lower bound increases for almost all the instances. A last remark we can give is that a very good feasible solution could be found in 1 or 2 hours. For example, for the instance with 12 nodes the gap is 5,62% (resp. 1,68%) after 1 hour (resp. 2 hours) with the path formulation for the MCSIPND$_s$ problem.

Finally, we present a little real french instance with 10 nodes, $|\mathcal{F}| = 25$ and $|K| = 10$. Figure 8 presents the optical network and the set of commodities.



Optical network Commodities

FIG. 8 – A real french instance with 10 nodes, $|\mathcal{F}| = 25$ and 10 commodities



MCSIPND$_m$ problem MCSIPND$_s$ problem

FIG. 9 – IP network solutions

Figure 9 gives the solutions for the two variants of the problem. The dashed lines represent the edges with capacity 2.5 Gbits and the solid ones the edges with capacity 10 Gbits. For the MCSIPND$_m$ problem, the number 2 on some edges indicates that we have installed two links between the towns, extremities of the edge. The optimal solutions shown in Figure 9 have been obtained with the Branch-and-Cut-and-Price algorithm based on the path formulation in 43 minutes for the multiple version and in only 10 seconds for the simple one.

7 Concluding remarks

In this paper we have considered the multilayer survivable network design problem which has applications to the design of reliable IP-over-optical network. We have considered the capacity dimensioning of the network. We have proposed two integer programming formulations for each of the two variants of the problem, simple and multiple. We have identified some valid inequalities, and described necessary conditions and sufficient conditions for a class of inequalities to define facets. Using this, we have developed Branch-and-Cut and Branch-and-Cut-and-Price algorithms for the problems and presented extensive computational results. These ones show that the path formulation based algorithm performs better than the one based on the node-arc formulation. The experimental results also show the effectiveness of the capacity demand cut, the cut-cycle and the star-partition inequalities for the problem.

Other variants of the multilayer network design problem are of interest for telecommunication operators and merit to be investigated. In particular those in which IP and optical layers should be treated simultaneously. This is our direction of futur research.

Also a more general variant of the problem, which merits to be studied, is when bounds are considered on the paths of the IP network. This issue has been treated in the literature in the monolayer case [12, 15, 22, 23]. An other interesting question would be to consider integer flows. This concept has also been investigated for the monolayer networks but without considering survivability [6, 10].

Références

[1] Abacus. http ://www.informatik.uni-koeln.de/abacus/.

[2] Ahuja R. K., Magnanti T. L., Orlin J.B. (1993). "Network Flows : Theory, Algorithms, and Applications". *Prentice Hall, Englewood Cliffs, NJ.*

[3] Barahona F., Mahjoub A. R. (1995). "On two-connected subgraph polytopes". *Discrete Mathematics 147*, 19-34.

[4] Barahona F. (1996). "Network Design Using Cut Inequalities". *SIAM Journal of Optimization 6*, 823-837.

[5] Barnhart C., Johnson E. L., Nemhauser G. L., Savelsbergh M. W. P., Vance P. H. (1998). "Branch-and-Price : column generation for solving huge integer programs". *Operations Research 46*, 316-329.

[6] Barnhart C., Hane C. A., Vance P. H. (2000). "Using Branch-and-Price-and-Cut to solve Origin-Destination Integer Multicommodity Flow Problem". *Operations Research 48*, 318-326.

[7] Borne S., Gourdin E., Liau B and Mahjoub A. R. (2006). "Design of survivable IP-over-Optical networks". *Annals of Operation Research 146*, 41-73.

[8] Borne S. (2006). "Sécurisation et dimensionnement de réseaux multicouches : modèles et polyèdres". PhD thesis, University Blaise Pascal, Clermont-Ferrand, France.

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

[10] Brunetta L., Conforti M., Fischetti M. (2000). "A polyhedral approach to an integer multicommodity flow problem". *Discrete Applied Mathematics 101*, 13-36.

[11] Cplex. http ://www.ilog.fr/products/cplex/.

[12] Dahl G., Huygens D., Mahjoub A. R., Pesneau P. (2006). "On the k-edge-disjoint 2-hop-constrained paths polytope". *Operations Research Letters 34*, 577-582.

[13] Dahl G., Stoer M. (1998). "A Cutting Plane Algorithm for Multicommodity Survivable Network Design Problems", *Journal on Computing 10*, 1-11.

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

[15] Fortz B., McCormick T., Mahjoub A. R., Pesneau P. (2006). "Two-edge connected subgraphs with bounded rings : Polyhedral results and Branch-and-Cut". *Mathematical Programming 105*, 85-111.

[16] Goldberg A. V., Tarjan R. E. (1988). "A new approach to the maximum-flow problem". *Journal of the Association for Computing Machinery 35*, 921-940.

[17] Gomory R. E., Hu T. C. (1961). "Multi-terminal network flows". *SIAM Journal on Applied Mathematics 9*, 551-570.

[18] Gouveia L., Patrício P., de Sousa A. , Valadas R. (2003). "MPLS over WDM Network Design with Packet Level QoS Constraints based on ILP Models ". *Proceedings IEEE INFOCOM 2003*, San Francisco.

[19] Gouveia L., Patrício P., de Sousa A. (2008). "Hop-Constrained Node Survivable Network Design : An Application to MPLS over WDM ". *Networks and Spatial Economics 8*, 3-21.

[20] Grötschel M., Monma C. L., Stoer M. (1995). "Design of survivable networks". Handbooks in OR & MS 7, *M.O. Ball et al. (Eds)*, 617-671.

[21] Gusfield D. (1990). "Very simple methods for all pairs network flow analysis". *SIAM Journal on Computing 19*, 143-155.

[22] Huygens D., Labbé M., Mahjoub A. R., Pesneau P. (2007). "Two Edge-Disjoint Hop-Constrained Paths : Valid Inequalities and Branch-and-Cut". *Networks 49*, 116-133.

[23] Huygens D., Mahjoub A. R., Pesneau P. (2004). "Two edge hop-constrained paths and polyhedra". *SIAM Journal on Discrete Mathematics 18*, 287-312.

[24] Jensen T. (2001). "Internet Protocol and Transport Protocols ". *Telektronikk 97*, 20-38.

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

[26] Kerivin H., Mahjoub A. R. (2005). "Design of Survivable Networks : A survey". *Networks 46*, 1-21.

[27] Knippel A., Lardeux B. (2007). "The Multi-Layered Network Design Problem". *European Journal of Operations Research 183*, 87-99.

[28] Lardeux B. (2005). "Conception de réseaux de Télécommunications Multicouches et Evolutifs". PhD thesis, Université Technologique de Compiègne, France.

[29] Lübbecke M. E. and Desrosiers J. (2002). "Selected topics in column generation". *Technical Report G-2002-64, GERAD.*

[30] Magnanti T. L., Mirchandani P., Vachani R. (1993). "The convex hull of two core capacitated network design problems". *Mathematical Programming 60*, 233-250.

[31] Magnanti T. L., Mirchandani P., Vachani R. (1995). "Modeling and solving the two-facility capacitated network loading problem". *Operations Research 43*, 142-157.

[32] Mahjoub A. R. (1994). "Two-edge connected spanning subgraphs and polyhedra". *Mathematical Programming 64*, 199-208.

[33] Orlowski S., Koster A. M. C. A., Raak C., Wessäly. R. (2006). "Two-layer Network Design by Branch-and-Cut featuring MIP-based Heuristics." *ZIP report, ZR-06-47.*

[34] Orlowski S., Wessäly. R. (2004). "An Interger Programming Model for Multi-Layer Network Design." *ZIP report, ZR-04-49.*

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

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

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

[38] Reinelt G. (1991). "TSPLIB - a traveling salesman problem library". *ORSA Journal on Computing 3*, 376-384.

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

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

[41] Stoer M. (1992). "Design of survivable networks". *Lectures Notes in Mathematics 1531*, Springer-Verlag.

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

[43] Voge M. E. (2006). "Optimisation des réseaux de télécommunications : Réseaux multiniveaux, Tolérance aux pannes et surveillance de traffic". PhD thesis, Université de Sophia-Antipolis, France.

[44] Zouganeli E. (2001). "Optical Network Functionality : From “Dumb Fat Pipes” to Bright Networking ". *Telektronikk 97*, 346-354.

Appendix : Proof of Theorem 4.9

Proof. We use ideas similar to those developed in [31]. Let us denote inequality (33) by

$$a^1x^1 + a^2x^2 \quad \geq \quad \alpha, \tag{44}$$

and let

$$b^1x^1 + b^2x^2 + \lambda f \quad \geq \quad \beta, \tag{45}$$

be a facet defining inequality of $\text{MCSIPND}_m^{na}(G, \mathcal{F}, K)$ such that the face defined by (44) is contained in that defined by (45). Let $L = \{(x^1, x^2, f) \in \text{MCSIPND}_m^{na}(G, \mathcal{F}, K) \,|\, a^1x^1 + a^2x^2 = \alpha\}$.

We first construct a feasible solution which satisfies (44) with equality. For each commodity k such that o_k and d_k are in W, as $G(W)$ is $\mathcal{F}$-connected, for all $j \in \{1, \ldots, t\}$, there exists a path P_j^k fully contained in $G_j(W)$, which connects o_k and d_k. If we install $\lceil \frac{\omega_k}{10} \rceil$ high capacities on each edge which belongs to at least one path P_j^k, then we can send a flow of ω_k along the path P_j^k from o_k to d_k for each failure j. By adding successively, in a similar way, the necessary capacity for all commodities, we obtain a feasible dimensioning for the edges of $E(W)$ according to the commodities of W. For the edges in $E(\overline{W})$, we similarly associate, for each commodity k such that o_k and $d_k \in \overline{W}$ and for each $j \in \{1, \ldots, t\}$, a path $\overline{P}_j^k$ between o_k and d_k. We install capacities on these paths in a similar way as in $G(W)$.

Now, consider an edge u_iv_i of $\delta_{G_i}(W)$ such that $u_i \in W$ and $v_i \in \overline{W}$.
Let $I_i = \{j \in \{1, \ldots, t\} \backslash \{i\} \,|\, u_iv_i \in F_j\}$. Then for all $j \in I_i$, there exists an edge, say u_jv_j, in $\delta(W) \cap F_i$ such that $u_jv_j \notin F_j$. In fact, if this is not the case, then there would exist $j \in I_i$ such that $\delta(W) \cap F_i \subseteq \delta(W) \cap F_j$. As by definition of I_i, $u_iv_i \in F_j \backslash F_i$, we would have $\delta(W) \cap F_i \subset F_j$. But this is a contradiction with Condition 3) of Theorem 4.8 (see Figure 10).

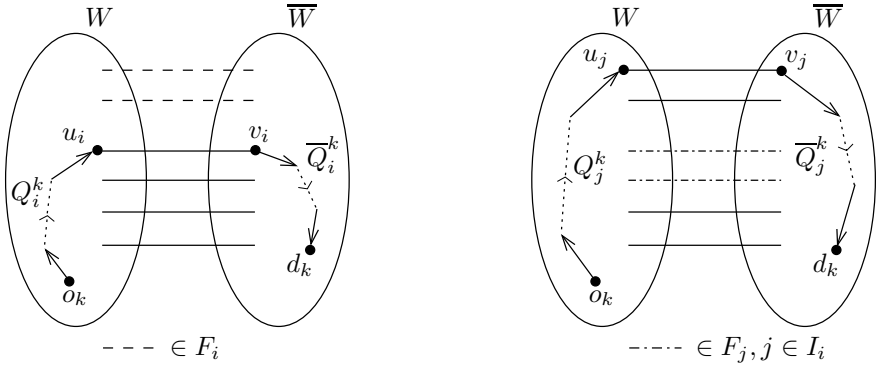


FIG. 10 –

Let $k \in \gamma(W)$ be a commodity accross the cut. As the graph $G(W)$ (resp. $G(\overline{W})$) is $\mathcal{F}$-connected, for all $j \in \{1, \ldots, t\} \backslash I_i$, there exists a path Q_j^k (resp. $\overline{Q}_j^k$) between o_k and u_i (resp. v_i and d_k) in $G_j(W)$ (resp. $\overline{G}_j(W)$). For $j \in I_i$, by the previous remark, there is an edge u_jv_j of $(\delta(W) \cap F_i) \backslash F_j$. We may suppose that $u_j \in W$ and $v_j \in \overline{W}$. For the commodity k, and for all $j \in I_i$, similarly, there is a path Q_j^k (resp. $\overline{Q}_j^k$) between o_k and u_j (resp. v_j and d_k) in $G_j(W)$ (resp. $\overline{G}_j(W)$).

Now, we can complete the partial dimensioning already performed in W and $\overline{W}$. We install for commodity k, $\lceil \frac{\omega_k}{10} \rceil$ high capacities on each edge that belongs to at least one path Q_j^k, $\overline{Q}_j^k$, $j = 1 \ldots, t$. We then add successively these capacities for each commodity of $\gamma(W)$. These new capacities, added to those already installed in W and $\overline{W}$, permit to have a feasible dimensioning for all the commodities and all the edges of $E(W) \cup E(\overline{W})$. This dimensioning of the edges of

$E(W) \cup E(\overline{W})$ can be given as follows,

$$\begin{array}{lll}
x^1_{uv} = 0 & \text{for all } uv \in E(W) \cup E(\overline{W}), & \\
x^2_{uv} = \displaystyle\sum_{\substack{k \,|\, o_k, d_k \in W \\ uv \in \cup_j P^k_j}} \left\lceil \frac{\omega_k}{10} \right\rceil + \sum_{\substack{k \in \gamma(W) \\ uv \in \cup_j Q^k_j}} \left\lceil \frac{\omega_k}{10} \right\rceil & \text{for all } uv \in E(W), & \\
x^2_{uv} = \displaystyle\sum_{\substack{k \,|\, o_k, d_k \in W \\ uv \in \cup_j \overline{P}^k_j}} \left\lceil \frac{\omega_k}{10} \right\rceil + \sum_{\substack{k \in \gamma(W) \\ uv \in \cup_j \overline{Q}^k_j}} \left\lceil \frac{\omega_k}{10} \right\rceil & \text{for all } uv \in E(\overline{W}). &
\end{array} \tag{46}$$

As $D_W \geq 4$, for the edges of $\delta(W)$, we can consider the following dimensioning,

$$\begin{array}{lcll}
x^1_{u_j v_j} & = & D_W \bmod 4 & \text{for all } j \in I_i \cup \{i\}, \\
x^1_{uv} & = & 0 & \text{for all } uv \in \delta(W) \setminus \cup_{j \in I_i \cup \{i\}} \{u_j v_j\}, \\
x^2_{u_j v_j} & = & \left\lfloor \frac{D_W}{4} \right\rfloor & \text{for all } j \in I_i \cup \{i\}, \\
x^2_{uv} & = & 0 & \text{for all } uv \in \delta(W) \setminus \cup_{j \in I_i \cup \{i\}} \{u_j v_j\}.
\end{array} \tag{47}$$

Also consider the flows on the paths P^k_j, $\overline{P}^k_j$, Q^k_j, $\overline{Q}^k_j$, $j = 1, \ldots, t$, given by

$$\begin{array}{ll}
f^{k,i}_{uv} = \omega_k & \text{for all } k \text{ such that } o_k, d_k \in W \text{ (resp. } o_k, d_k \in \overline{W}), \\
& j = 1, \ldots, t,\ uv \in P^k_j, \text{ (resp. } uv \in \overline{P}^k_j) \text{ and } uv \text{ traversed from } u \\
& \text{to } v \text{ in } P^k_j \text{ (resp. } \overline{P}^k_j), \\
f^{k,i}_{uv} = \omega_k & \text{for all } k \in \gamma(W)\, j = 1, \ldots, t,\ uv \in Q^k_j, \text{ (resp. } uv \in \overline{Q}^k_j) \\
& \text{and } uv \text{ traversed from } u \text{ to } v \text{ in } Q^k_j \text{ (resp. } \overline{Q}^k_j), \\
f^{k,i}_{uv} = \omega_k & \text{for all } k \in \gamma^+(W) \text{ (resp. } \gamma^-(W)), \\
& u = u_j \,(v_j), v = v_j \,(u_j)\, j \in I_i \cup \{i\}, \\
f^{k,i}_{uv} = 0 & \text{otherwise.}
\end{array} \tag{48}$$

The solution (x^1, x^2, f) given by (46), (47) and (48) is feasible for $\text{MCSIPND}^{na}_m(G, \mathcal{F}, K)$.

Let $pq \in E(W) \cup E(\overline{W}) \cup (F_i \cap \delta(W))$. And let $(x^{1'}, x^{2'}, f')$ be the solution such that

$$\begin{array}{lcll}
x^{1'}_{uv} & = & x^1_{uv} & \text{for all } uv \neq pq, \\
x^{1'}_{pq} & = & x^1_{pq} + 1, & \\
x^{2'} & = & x^2, & \\
f' & = & f. &
\end{array}$$

Clearly, $(x^{1'}, x^{2'}, f')$ is feasible for $\text{MCSIPND}^{na}_m(G, \mathcal{F}, K)$. Furthermore, (x^1, x^2, f) and $(x^{1'}, x^{2'}, f')$ satisfy inequality (44) with equality. In consequences, both solutions satisfy inequality (45) with equality. This yields $b^1_{pq} = 0$. Similarly, one can show that $b^2_{pq} = 0$. As pq is an arbitrary edge of $E(W) \cup E(\overline{W}) \cup (F_i \cap \delta(W))$, we then have $b^1_{uv} = b^2_{uv} = 0$ for all $uv \in E(W) \cup E(\overline{W}) \cup (F_i \cap \delta(W))$.

Now, consider the edge $u_i v_i$ (introduced above) and the solution $(\tilde{x}^1, \tilde{x}^2, \tilde{f})$ such that

$$\begin{array}{lcll}
\tilde{x}^1_{u_i v_i} & = & x^1_{u_i v_i} + 4, & \\
\tilde{x}^1_{uv} & = & x^1_{uv} & \text{for all } uv \in E \text{ such that } uv \neq u_i v_i, \\
\tilde{x}^2_{u_i v_i} & = & x^2_{u_i v_i} - 1, & \\
\tilde{x}^2_{uv} & = & x^2_{uv} & \text{for all } uv \in E \text{ such that } uv \neq u_i v_i, \\
\tilde{f} & = & f. &
\end{array}$$

The solution $(\tilde{x}^1, \tilde{x}^2, \tilde{f})$ is defined from (x^1, x^2, f) by replacing a high capacity on $u_i v_i$ by 4 small ones. Hence, this solution is feasible. Furthermore, as $(\tilde{x}^1, \tilde{x}^2, \tilde{f})$ satisfies (44) with equality, it also

satisfies (45) with equality. This implies that $4b^1_{u_iv_i} = b^2_{u_iv_i}$. As u_iv_i is an arbitrary edge of $\delta_{G_i}(W)$, we then get

$$b^2_{uv} = 4b^1_{uv} \text{ for all } uv \in \delta(W)\backslash F_i.$$

Consider again an edge $pq \in E(W) \cup E(\overline{W}) \cup F_i$ and the solution (x^1, x^2, f) introduced above. Let $k' \in K$ be a commodity and $j' \in \{1, \ldots, t\}$ a failure. Consider $(\hat{x}^1, \hat{x}^2, \hat{f})$ given by

$$\begin{array}{rcl}
\hat{x}^1 & = & x^1, \\
\hat{x}^2 & = & x^2, \\
\hat{f}^{k',j'}_{pq} & = & f^{k',j'}_{pq} + \varepsilon, \\
\hat{f}^{k',j'}_{qp} & = & f^{k',j'}_{qp} + \varepsilon, \\
\hat{f}^{k,j}_{uv} & = & f^{k,j}_{uv} \text{ for all } uv \in E\backslash\{pq\}, \text{ for all } k \in K, \text{ for all } j \in \{1, \ldots, t\},
\end{array}$$

where $0 < \varepsilon < \frac{1}{2}$. Note that as the capacity of edge pq is sufficiently big, we have a residual capacity to carry more flow. In consequences, $(\hat{x}^1, \hat{x}^2, \hat{f})$ is feasible. Moreover, this solution satisfies inequality (44), and hence (45), with equality. As (x^1, x^2, f) also satisfies (45) with equality, we obtain $\lambda^{k',j'}_{pq}\varepsilon + \lambda^{k',j'}_{qp}\varepsilon = 0$ implying that $\lambda^{k',j'}_{pq} = -\lambda^{k',j'}_{qp}$. As pq is an arbitrary edge of $E(W) \cup E(\overline{W}) \cup F_i$, and k' and j' are arbitrary in K and $\{1, \ldots, t\}$, respectively, we obtain

$$\lambda^{k,j}_{uv} = -\lambda^{k,j}_{vu} \text{ for all } uv \in E(W) \cup E(\overline{W}) \cup F_i, \text{ for all } k \in K \text{ and all } j \in \{1, \ldots, t\}.$$

Now consider again the edge u_iv_i of $\delta(W)\backslash F_i$. By Condition 5) of Theorem 4.8, if we consider the solution (x^1, x^2, f), as u_iv_i has a sufficiently big capacity, we can add more flow on this edge. In a similar way we can show that $\lambda^{k',j'}_{u_iv_i} = -\lambda^{k',j'}_{v_iu_i}$. And as u_iv_i is chosen arbitrarly in $\delta_{G_i}(W)$, like k' and j' in K and $\{1, \ldots, t\}$, we obtain

$$\lambda^{k,j}_{uv} = -\lambda^{k,j}_{vu} \text{ for all } uv \in \delta_{G_i}(W), \text{ for all } k \in K \text{ and all } j \in \{1, \ldots, t\}.$$

We have then shown that

$$\lambda^{k,i}_{uv} = -\lambda^{k,i}_{vu} \text{ for all } uv \in E, \text{ for all } k \in K \text{ and all } j \in \{1, \ldots, t\}. \tag{49}$$

We now show that $\sum_{k\in K}\sum_{uv\in E}(\lambda^{k,j}_{uv} f^{k,j}_{uv} + \lambda^{k,j}_{vu} f^{k,j}_{vu})$ is a constant for all failure $j = 1, \ldots, t$, by showing that the sum of the coefficients, corresponding to any cycle in the network, equals zero. Let Δ denote the set of cycles in $D = (V, A)$. Consider a failure $j \in \{1, \ldots, t\}$ and a particular cycle $\xi \in \Delta$. Let $\lambda^{k,j}_{\xi} = \sum_{(u,v)\in\xi} \lambda^{k,j}_{uv}$. We will show that $\lambda^{k,j}_{\xi} = 0$ for all cycle $\xi \in \Delta$. Call ξ an *s-intersection cycle* with respect to the cut $\delta_G(W)$ if ξ contains exactly s arcs of $\delta_G(W)$. Note that s must be even as ξ is a cycle.

If ξ is a 0-intersection cycle, then ξ is completely contained in W or $\overline{W}$. We suppose w.l.o.g., that ξ is in W. Let $k \in K$ and $(\check{x}^1, \check{x}^2, \check{f})$ be the solution given by

$$\begin{array}{rcll}
\check{x}^1_{uv} & = & x^1_{uv} + 1 & \text{for all } uv \in E(W) \text{ such that } (u,v) \text{ ou } (v,u) \in \xi, \\
\check{x}^1_{uv} & = & x^1_{uv} & \text{for all } uv \in E(W) \text{ such that } (u,v) \text{ ou } (v,u) \notin \xi, \\
\check{x}^2_{uv} & = & x^2_{uv} & \text{for all } uv \in E, \\
\check{f}^{k,j}_{uv} & = & f^{k,j}_{uv} + 1 & \text{for all } (u,v) \in \xi, \\
\check{f}^{k,j}_{uv} & = & f^{k,j}_{uv} & \text{for all } (u,v) \notin \xi, \\
\check{f}^{k,h}_{uv} & = & f^{k,h}_{uv} & \text{for all } (u,v) \in A, \text{ for all } h \in \{1, \ldots, t\}\backslash\{j\}.
\end{array}$$

Here, we construct solution $(\check{x}^1, \check{x}^2, \check{f})$ from solution (x^1, x^2, f) by installing a small capacity on all the edges in ξ and by sending an additional unit of flow on ξ. Solution $(\check{x}^1, \check{x}^2, \check{f})$ is still feasible and satisfies (44) with equality. Hence $(\check{x}^1, \check{x}^2, \check{f})$ satisfies (45) with equality, and in consequence we get

$$\sum_{uv\in E(W)} b^1_{uv} x^1_{uv} - \sum_{uv\in E(W)} b^1_{uv} \check{x}^1_{uv} + \sum_{(u,v)\in\xi} (\lambda^{k,j}_{uv} f^{k,j}_{uv} - \lambda^{k,j}_{vu} f^{k,j}_{vu}) = 0.$$

As $b^1_{uv} = 0$ for all $uv \in E(W)$, we obtain $\sum_{(u,v)\in\xi} -\lambda^{k,j}_{uv} = 0$, and then $\lambda^{k,j}_{\xi} = 0$. As cycle ξ, the commodity k and the failure j are arbitrary, we have then shown that

$$\begin{array}{ll} \lambda^{k,j}_{\xi} = 0 & \text{for all 0-intersection cycle } \xi \in \Delta, \text{ for all } k \in K \\ & \text{and for all } j \in \{1,\ldots,t\}. \end{array} \tag{50}$$

Now suppose that ξ is a 2-intersection cycle. Assume that $(p,\bar{p})$ and $(q,\bar{q})$ are the cutset arcs belonging to $\delta_G(W) \cap \xi$ with $p,q \in W$ and $\bar{p},\bar{q} \in \overline{W}$.
If $p = q$ and $\bar{p} = \bar{q}$, then ξ can be decomposed into three cycles : a 0-intersection cycle denoted by ξ_1 contained in W, a 0-intersection cycle denoted by ξ_2 contained in $\overline{W}$ and a cycle formed by the arcs $(p,\bar{p})$ and $(\bar{p},p) = (\bar{q},q)$. We then have $\lambda^{k,j}_{\xi} = \lambda^{k,j}_{\xi_1} + \lambda^{k,j}_{\xi_2} + \lambda^{k,j}_{p\bar{p}} + \lambda^{k,j}_{\bar{p}p}$. As ξ_1 and ξ_2 are 0-intersection cycles, $\lambda^{k,j}_{\xi_1} = \lambda^{k,j}_{\xi_2} = 0$ and by (49), we have $\lambda^{k,j}_{p\bar{p}} = -\lambda^{k,j}_{\bar{p}p}$. Thus $\lambda^{k,j}_{\xi} = 0$.
Now suppose that $p\bar{p} \neq q\bar{q}$ and one of the two edges $p\bar{p}$ or $q\bar{q}$ belongs to F_i. We suppose, w.l.o.g., that $q\bar{q} \in F_i$ and $p\bar{p} \notin F_i$. Let $p\bar{p}$ be the edge which belongs to $\delta_{G_i}(W)$. One can construct a feasible solution $(\breve{x}^1, \breve{x}^2, \breve{f})$ in a similar way as (x^1, x^2, f) (given above for the edge u_iv_i). Remark that $\breve{x}^1_{p\bar{p}} = D_W \mod 4$ and $\breve{x}^2_{p\bar{p}} = \left\lfloor \frac{D_W}{4} \right\rfloor$. Consider the solution obtained from $(\breve{x}^1, \breve{x}^2, \breve{f})$ by installing a small capacity on all the edges of $\xi \cap (E(W) \cup E(\overline{W}) \cup F_i)$, and by sending ε units of flow on ξ (for a certain $\varepsilon > 0$). The additional flow is possible since by Condition 5) of Theorem 4.8, there is a positive residual capacity on the edge $p\bar{p}$. This new solution is feasible. As both solutions satisfy (44) with equality and hence (45), we get $\lambda^{k,j}_{\xi} = 0$.
If $p\bar{p} = q\bar{q}$ and $p\bar{p}, q\bar{q} \notin F_i$, we define a solution $(\dot{x}^1, \dot{x}^2, \dot{f})$ by considering for the edges of the cut $\delta_{G_i}(W)$ the following values

$$\begin{array}{lcll} \dot{x}^1_{p\bar{p}} & = & D_W \mod 4, & \\ \dot{x}^1_{uv} & = & 0 & \text{for all } uv \in \delta_{G_i}(W), \\ \dot{x}^2_{p\bar{p}} & = & \left\lfloor \frac{D_W}{4} \right\rfloor - 1, & \\ \dot{x}^2_{q\bar{q}} & = & 1, & \\ \dot{x}^2_{uv} & = & 0 & \text{for all } uv \in \delta_{G_i}(W)\backslash\{p\bar{p}, q\bar{q}\}. \end{array}$$

For the dimensioning of $E(W)$ and $E(\overline{W})$, we suppose that we have installed a sufficient capacity which permits to carry the flows of the commodities in W and $\overline{W}$ on the paths fully contained in $G(W)$ and $G(\overline{W})$. These capacities can be taken as big as we want. So, by Condition 5) of Theorem 4.8, the commodities of $\gamma(W)$ can be routed in such a way that $p\bar{p}$ and $q\bar{q}$ contain at least ε (for a certain $\varepsilon>0$) units of residual capacity. This solution is feasible and satisfies (44) and then (45) with equality. Using the residual capacity ε on $p\bar{p}$ and $q\bar{q}$, one can obtain a new feasible solution from $(\dot{x}^1, \dot{x}^2, \dot{f})$ by adding an additional flow ε along ξ for commodity k, and installing a small capacity on the edges of ξ not in $\delta_{G_i}(W)$. This new solution is also feasible and satisfies (44) and (45) as equalities. As commodity $k \in K$, the failure $j \in \{1,\ldots,t\}$ and the 2-intersection cycle $\xi \in \Delta$ are arbitrary, we have that

$$\begin{array}{ll} \lambda^{k,j}_{\xi} = 0 & \text{for all 2-intersection cycle } \xi \in \Delta, \text{ for all } k \in K \\ & \text{and for all } j \in \{1,\ldots,t\}. \end{array} \tag{51}$$

Now consider an arbitrary s-intersection cycle ξ. Let k be a commodity and j be a failure. Let ξ be the cycle given by $\{(s_1,s_2),(s_2,s_3),\ldots,(s_T,s_1)\}$ with $s_1 \in W$. Let (s_{t_1}, s_{t_2}) be the first arc of the cycle ξ that crosses $\delta(W)$ and (s_{t_3}, s_{t_4}) the first subsequent arc that re-enters set W. Note that cycle $\xi' = \{(s_{t_1},s_{t_2}),\ldots,(s_{t_3},s_{t_4}),(s_{t_4},s_{t_1})\}$ is a 2-intersection cycle. We then have $\lambda^{k,j}_{\xi'\backslash(s_{t_1},s_{t_4})} = \lambda^{k,j}_{s_{t_1}s_{t_4}}$. So, we can replace the path $\xi'\backslash(s_{t_1},s_{t_4})$ by the arc (s_{t_1},s_{t_4}). Repeating this argument, one can construct a 0-intersection cycle ψ that satisfies $\lambda^{k,j}_{\xi} = \lambda^{k,j}_{\psi}$. As k and j are arbitrary, we obtain that

$$\begin{array}{ll} \lambda^{k,j}_{\xi} = 0 & \text{for all cycle } \xi \in \Delta, \text{ for all } k \in K \\ & \text{and for all } j \in \{1,\ldots,t\}. \end{array} \tag{52}$$

We have shown that $\sum_{k\in K}\sum_{uv\in E}(\lambda_{uv}^{k,j} f_{uv}^{k,j} + \lambda_{vu}^{k,j} f_{vu}^{k,j})$ is a constant, say λ_j, for any failure $j \in \{1,\dots,t\}$.

We can now show that $b_{uv}^1 = \rho^1$ and $b_{uv}^2 = \rho^2$ for all $uv \in \delta_{G_i}(W)$. Consider the solution (x^1, x^2, f) given in the beginning of the proof, an edge $pq \in \delta_{G_i}(W)$ such that $pq \neq u_iv_i$, a commodity $k' \in K$ and a failure $j \in \{1,\dots,t\}$. Let $P(u_i,p)$ be a path from node u_i to p fully contained in W and $P(q,v_i)$ a path from q to v_i in $\overline{W}$. Consider the solution $(\ddot{x}^1, \ddot{x}^2, \ddot{f})$ such that

$$\begin{array}{rcll}
\ddot{x}^1_{u_iv_i} & = & x^1_{u_iv_i} - 1 & = \ D_W \bmod 4 - 1, \\
\ddot{x}^1_{pq} & = & 1, & \\
\ddot{x}^1_{uv} & = & x^1_{uv} & \text{for all } uv \in \delta_G(W)\backslash\{u_iv_i, pq\}, \\
\ddot{x}^1_{uv} & = & x^1_{uv} + 1 & \text{for all } (u,v) \in P(u_i,p) \cup P(q,v_i), \\
\ddot{x}^1_{uv} & = & x^1_{uv} & \text{for all } (u,v) \in E(W) \cup E(\overline{W})\backslash(P(u_i,p) \cup P(q,v_i)), \\
\ddot{x}^2_{uv} & = & x^2_{uv} & \text{for all } uv \in E.
\end{array}$$

Define the flows as follows.

$$\begin{array}{rcl}
\ddot{f}^{k',j}_{uv} & = & f^{k',j}_{uv} + 1 \quad \text{for all } (u,v) \in P(u_i,p) \cup P(q,v_i), \\
\ddot{f}^{k',j}_{u_iv_i} & = & f^{k',j}_{u_iv_i} - 1, \\
\ddot{f}^{k',j}_{pq} & = & 1, \\
\ddot{f}^{k',j}_{uv} & = & f^{k',j}_{uv} \quad \text{for all } uv \in E\backslash(P(u_i,p) \cup P(q,v_i) \cup \{u_iv_i, pq\}), \\
\ddot{f}^{k,h}_{uv} & = & f^{k,h}_{uv} \quad \text{for all } uv \in E, \text{ for all } k \in K\backslash\{k'\}, \text{ for all } h \in \{1,\dots,t\}\backslash\{j\}.
\end{array}$$

Solutions (x^1, x^2, f) and $(\ddot{x}^1, \ddot{x}^2, \ddot{f})$ are feasible and satisfy constraint (44) with equality. In consequence, they also satisfy (45) with equality. This implies that

$$b^1_{u_iv_i} - b^1_{pq} - \sum_{uv\in P(u_i,p)\cup P(q,v_i)} \rho^1_{uv} - \sum_{(u,v)\in P(u_i,p)} \lambda^{k',j}_{uv} - \sum_{(u,v)\in P(q,v_i)} \lambda^{k',j}_{uv} + \lambda^{k',j}_{u_iv_i} - \lambda^{k',j}_{pq} = 0.$$

As $b^1_{uv} = 0$ for all $uv \in E(W) \cup E(\overline{w})$ and by (49) $\lambda^{k',j}_{u_iv_i} = -\lambda^{k',j}_{v_iu_i}$, we have

$$b^1_{u_iv_i} - b^1_{pq} - \sum_{(u,v)\in P(u_i,p)} \lambda^{k',j}_{uv} - \lambda^{k',j}_{pq} - \sum_{(u,v)\in P(q,v_i)} \lambda^{k',j}_{uv} - \lambda^{k',j}_{v_iu_i} = 0.$$

As $P(u_i,p)$, pq, $P(q,v_i)$ and v_iu_i form a cycle, we have $b^1_{u_iv_i} - b^1_{pq} = 0$. As pq was chosen arbitrarily, we obtain $b^1_{uv} = \rho^1$ for all $uv \in \delta_{G_i}(W)$ for $\rho^1 \in I\!R$. Since $b^2_{uv} = 4b^1_{uv}$ for all $uv \in \delta_{G_i}(W)$, we have also $b^2_{uv} = 4\rho^1$ for all $uv \in \delta_{G_i}(W)$.

Thus, inequality (45) is equivalent to

$$\rho^1 x^1(\delta_{G_i}(W)) + 4\rho^1 x^2(\delta_{G_i}(W)) + \sum_{j=1}^{t} \lambda_j = \lambda,$$

which implies that

$$\rho^1 x^1(\delta_{G_i}(W)) + 4\rho^1 x^2(\delta_{G_i}(W)) = \lambda'$$

where $\lambda' = \lambda - \sum_{j=1}^{t} \lambda_j$. As the face defined by inequality (45) is not empty, $\rho^1 \neq 0$ and we obtain that

$$x^1(\delta_{G_i}(W)) + 4x^2(\delta_{G_i}(W)) = \frac{\lambda'}{\rho^1} = D_W.$$

□