RAIRO-Oper. Res. 48 (2014) 167–188
DOI: 10.1051/ro/2014006

RAIRO Operations Research
www.rairo-ro.org

A BRANCH-AND-CUT FOR THE NON-DISJOINT m-RING-STAR PROBLEM

Pierre Fouilhoux[1] and Aurélien Questel[1]

Abstract. In this article we study the realistic network topology of Synchronous Digital Hierarchy (SDH) networks. We describe how providers fulfill customer connectivity requirements. We show that SDH Network design reduces to the Non-Disjoint m-Ring-Star Problem (NDRSP). We first show that there is no two-index integer formulation for this problem. We then present a natural 3-index formulation for the NDRSP together with some classes of valid inequalities that are used as cutting planes in a Branch-and-Cut approach. We propose a polyhedral study of a polytope associated with this formulation. Finally, we present our Branch-and-Cut algorithm and give some experimental results on both random and real instances.

Keywords. Realistic SDH network, non-disjoint m-ring-star problem, polyhedral approach, branch-and-cut algorithm.

Mathematics Subject Classification. 90B10, 90C10, 90C57, 90C90.

1. Introduction

In order to propose a reliable service to their clients, providers need to embed particular topology for their urban optical network. In particular, SDH technology (Synchronous Digital Hierarchy) is a wide-spread protocol for transmitting digital data at high speed, corresponding to the SONET protocol in the United States. SDH topology can be described as a network which consists in a set of optical fiber rings where clients are nodes of a ring [15, 16]. Moreover, since an optical cable is composed by a lot of optical fibers and since a ring uses only one fiber, many rings

Received September 11, 2013. Accepted November 28, 2013.

[1] Laboratoire LIP6, Université Pierre et Marie Curie, 4 place Jussieu 75005 Paris, France.
pierre.fouilhoux@lip6.fr; aurelien.questel@lip6.fr

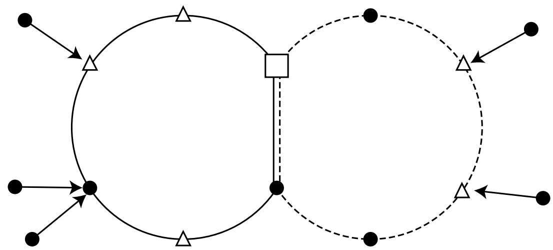


FIGURE 1. Two Non-Disjoint Ring-Stars.

share the same cable. In this article, we will show that designing a SDH topology can be reduced to finding a set of non-disjoint ring-stars. Given a mixed graph $G = (V, E, A)$, we call V the set of nodes which is partitioned into the *depot* 0, the set of *clients* U, and the set of *interconnection nodes* W, also known as *Steiner* nodes. We will denote $n = |V|$ and $n_u = |U|$. Each client $i \in U$ has a demand d_i. The set of arcs A is equal to $\cup_{i \in U} L_i$, where L_i is the set of potential assignments of client $i \in U$, $L_i \subset \{(i,j), j \in V \setminus \{0\}\}$. Every edge $e \in E$ corresponds to a cable which is made of γ_e optical fibers. To each arc $(i,j) \in A$, we associate an *assignment cost* $c_{ij} \in \mathbb{R}^+$, and to each edge $e \in E$ we associate a *routing cost* $l_e \in \mathbb{R}^+$.

A couple $R = (E_R, L_R)$ will be called a *ring-star* if E_R is an elementary cycle of E going through the depot 0 and L_R is a subset of arcs of A such that $\forall (i,j) \in L_R, j \in V_R$ where V_R is the set of nodes belonging to the cycle. U_R will be the set of clients served by R. Figure 1 shows two non-disjoint ring-stars, the first one in dashed lines, the other one in continuous lines. The clients are represented by circles, while the Steiner nodes are represented by triangles. The depot is represented by a square. We remark that the ring-stars share the vertical edge.

We will denote $d(R) = \sum_{i \in U(R)} d_i$ the demand served by R. Given $Q \in \mathbb{R}^+$, a ring-star R is called a Q-ring-star if $d(R) \leq Q$. Given $m \in \mathbb{N}^+$, the Non-disjoint (Steiner) m-Ring-Star Problem (NDRSP) consists in finding a set of m Q-ring-stars on a given graph G such that: each edge $e \in E$ belongs to at most γ_e rings; each client is assigned to a ring; and the sum of assignment and routing costs is minimized. Such an instance will be denoted $I = (G, d, \gamma, c, l, Q, m)$. In the sequel, we assume that the number m of rings of capacity Q is enough to carry out all the demands.

The NDRSP may seem similar in some aspects to vehicle routing problems, in particular to the well-known CVRP (see for instance [5, 6, 8, 13, 14]). An instance of the CVRP can be defined as a depot, a set of customers, each with a specific demand, and a capacity Q for the vehicles. A solution is a partition of the client set into subsets along with an order of visit on each subset, such that the total distance traveled is minimized. A classic trick is then to use a complete graph on the same node set, but where each edge represents the shortest path between two nodes. Once a solution is obtained, one can easily find the actual path of each vehicle. It is worth noticing that the topology of these pathways will not be constrained, and especially it will not necessarily be rings. On the contrary, for the NDRSP, the elementary ring topology must be verified in the original graph as it is essential to the reliability of an SDH network. Nevertheless, the trick of reducing the CVRP to the search of disjoint rings cannot therefore be applied to NDRSP. Moreover, in optical networks, cable capacity constraints has no equivalent in vehicle routing problems. In fact, the NDRSP seems to be closer to network topology problems.

To the best of our knowledge, there is no work about the NDRSP in the network design literature. The 1-Ring-Star case has been treated in [12] where an integer formulation has been used in a Branch-and-Cut framework. Another integer formulation together with a polyhedral study of the 1-Ring-Star can be found in [11]. We also note that if all the assignments are previously known and if $\gamma_e \geq m, \forall e \in E$, our problem reduces to m independant Steiner TSP [3]. Consequently, NDRSP is clearly NP-hard.

The disjoint case of NDRSP was introduced in [4]. In this article, the authors present and compare two integer formulations and propose new valid inequalities to strengthen the linear programming relaxation. They have implemented a Branch-and-Cut algorithm which has been able to solve instances up to 101 nodes. In [10], the authors develop a Branch-and-Cut-and-Price algorithm for the disjoint case where the edge weights satisfy the triangular inequality. The authors compare the performance of their algorithm in regards of the results in [4], improving the previous results on several instances.

This paper is organized as follows. In Section 2 we describe the customer connectivity requirements in real SDH networks. We then the show how an instance of our problem can be built from a real SDH network. Afterwards, in Section 3 we will discuss the encoding of a solution, showing there is no two-index formulation of our problem. In Section 4 we will present a natural 3-index formulation, and we will propose several new families of valid inequalities. In Section 5, we will study a full dimensional polytope containing the NDRSP polytope as a face. This polytope is defined as a relaxation of the fact that all clients must be served. In Section 6 we will present our Branch-and-Cut algorithm along with the separation of the constraints and a primal heuristic. Finally, in Section 7 we will present the experimental results on both random and real instances.

Let us introduce some additional notations. Let $G = (V, E, A)$ be a graph defined as above and $S \subset V$. $\bar{S} = \{i \in V : i \notin S\}$ is the complement of S. $\delta(S) = \{\{i, j\} \in E : i \in S, j \in \bar{S}\}$ is called the *cutset* defined by S, $\delta^+(S) = \{(i, j) \in A : i \in S, j \in \bar{S}\}$ is the *set of outgoing arcs* from S, and $\delta^-(S) = \{(i, j) \in A : i \in \bar{S}, j \in S\}$ is the *set of ingoing arcs* to S. If $S = \{i\}$, we simply write $\delta(i)$ instead of $\delta(\{i\})$.

2. Modeling

The strenght of SDH networks lies in the management, monitoring, alarm and self-healing functions. The main protection comes from the ring structure of the

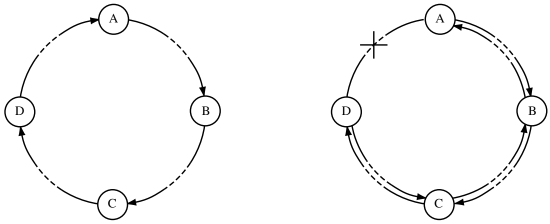


FIGURE 2. Rerouting in case of failure.

network. The data flows in one direction and if a connection is broken, the data is immediately redirected in the opposite direction (Fig. 2). Thus the failure of a link does not completely isolate one element.

These optical networks include four types of point: the *depot*, the *Network Flexibility Points* (NFP) and the *Building Facility Points* (BFP). The depot is a building from which the entire network is managed and linked to the rest of the worldwide network. The NFP are small rooms located under the sidewalks where the main cables are connected together. The BFP are the entry points into a building that contains several client equipments. These three types of point are connected with cables deployed in the sewers, pavement or in the underground corridors of the public transport network.

A SDH ring is a sequence of fiber segments going from the depot, passing through NFP, BFP and client equipments, and going back to the depot.

A client equipment is linked either to only one BFP (simple adduction) or to two different BFP (double adduction). The most common configuration is the *simple adduction*: a single cable enters the building through the BFP. Consequently, two fibers of the same cable must be used to guarantee the ring structure at the fiber level (Fig. 3a). We can see that this type of connection is not secure, because if the cable is disconnected then the site is offline. However, only the client would be disconnected, the remaining elements of the ring would still be online. In order to obtain more reliability, some clients may require a *double adduction*: two different cables enter the building through different paths, and the ring structure is checked even at the client site (Fig. 3b). In addition, each client has a particular bandwidth demand, and each ring has a fixed bandwidth capacity Q. Consequently the sum of the client demands of a ring must be Q or less.

An optical network is hence given by a depot 0, a set of clients U, a set of BFP, a set of NFP and the set of cables. Each cable e has a limited number γ_e of fibers and each client i has a bandwidth demand d_i. Finally, a ring has a maximal bandwidth capacity limited by Q. Given an optical network, the *SDH network design problem* consists in finding a set of SDH rings spanning all the clients satisfying the bandwidth and fiber capacities.

Figure 4a represents a simple but realistic example of SDH network with 6 clients, 7 BFP and 7 NFP. Figure 4b shows a solution of the SDH network design problem on this network. This solution is made of three rings, one in dotted lines

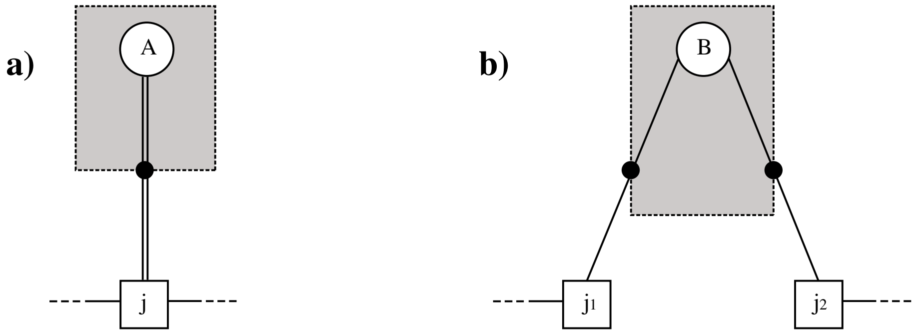


FIGURE 3. Simple and double adduction.

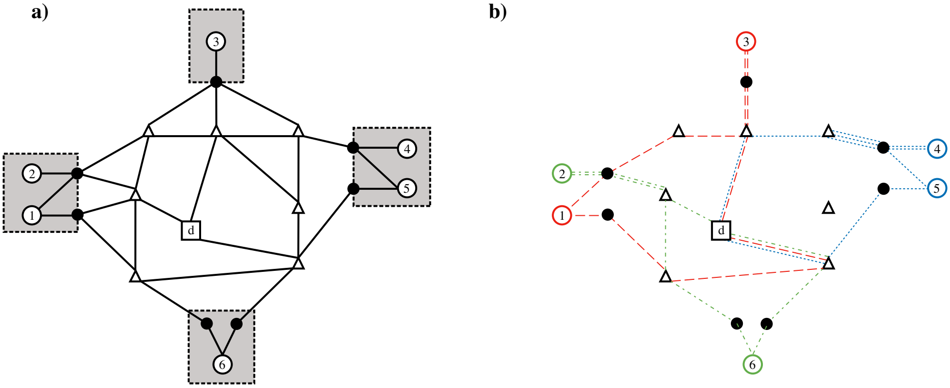


FIGURE 4. (a) A simple but realistic SDH network. (b) The fiber paths.

serving clients 4 and 5, one in dashed lines serving clients 1 and 3, and the last one serving the remaining clients 2 and 6. We note that some cables are used by several rings.

We can now show how to construct an instance of the NDRSP from such an optical network. The graph is constructed as follows:

- we create a node 0 for the depot;
- for each client we create a client node i;
- for each interconnection point we create a Steiner node;
- for each cable e linking two interconnection points or a connection point and the depot we create an edge e and we set l_e as the length of the cable;
- if a client i orders a double adduction, for each BFP linked to client i, we create a Steiner node j, an edge $e = \{i, j\}$ and we set l_e as the length of the corresponding cable. We also create an arc (i, i) and set $c_{ii} = 0$;

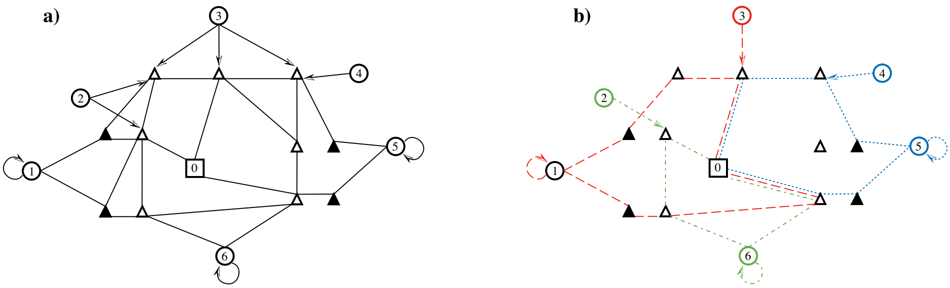


FIGURE 5. (a) A Graph model. (b) A solution.

- if a client orders a simple adduction, we create an arc (i, j) for each potential path going from the client to an interconnection point j linked to a BFP of client i. We set c_{ij} as $2 * l_p$, where p is the corresponding path.

By construction, each Q-ring-star R on graph G corresponds to a SDH ring on the optical network and conversely. Moreover, the total assignment and routing cost of R is exactly the total length of fiber used on the optical network. Hence, we can state the following result:

Theorem 2.1. *The SDH network design problem reduces to the NDRSP.*

Figure 5a (resp. 5b) shows the graph model corresponding to the network (resp. solution) of Figure 4.

3. Solution encoding

In order to derive integer formulations for the NDRSP, we need to determine a set of variables whose values lead in polynomial time to optimal solutions. For the disjoint case of our problem [4], the authors have proposed two-index formulations, that is to say, formulations based on integer variables $x_e, e \in E$, corresponding to the number of ring-stars using edge e. In this case the x_e variables are binary, implying that $\{e \in E : x_e = 1\}$ is immediatly a set of disjoint ring-stars. We will show that, for the non-disjoint case of the Ring-Star Problem, a solution cannot be described by such a vector.

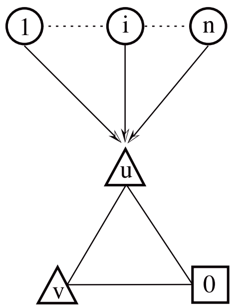
We consider the Ring-Star Decomposition Problem (RSDP) defined as follows.

(RSDP): Given an NDRSP instance $I = (G, d, \gamma, c, l, Q, m)$ and a vector $(x, y) \in I\!N^{|E|} \times \{0, 1\}^{|A|}$, is there a solution $R = (R_1, \ldots, R_m)$, such that:
- $R_i = (E(R_i), L(R_i)), i = 1, \ldots, m,$
- $|\{R_i, i = 1, \ldots, m : e \in E(R_i)\}| = x_e, \forall e \in E,$
- $\cup_{i=1,\ldots,m} L(R_i) = \{(i, j) \in A : y_{ij} = 1\}$?

Theorem 3.1. *RSDP is NP-complete.*

Proof. We show that the Bin Packing Problem (BPP) can be reduced to the RSDP. An instance of the BPP can be described as a finite set of items $U = \{1, \dots, n\}$, a size $a_i \in I\!N$ for each $i \in U$, a positive integer capacity Q, and a positive integer m. The BPP consists in answering the question: is there a partition of U into disjoint sets $U_1, \dots, U_m$ such that the sum of the sizes of the items in each U_i is Q or less?

Given such an instance $I = \{U, a, m, Q\}$ of BPP, it is possible to build an instance I' of RSDP as follows. We set $G = (V, E, A)$ as a graph with $V = \{0, u, v\} \cup U$ and $E = \{\{0, u\}, \{u, v\}, \{v, 0\}\}$. Let $d_i = a_i$ and $L_i = \{(i, u)\}$ be the demand and potential assignment of client $i = 1, \dots, m$. The following figure illustrates the RSDP graph built from instance I of the BPP.



Finally let $x_e = m, \forall e \in E$, and $y_{iu} = 1, \forall i \in U$. Hence, we can clearly see that the BPP instance I has a solution if and only if there is a solution on the RSDP instance I'. Since the BPP is NP-complete [9], we can state that the RSD is also NP-complete. □

The preceding result shows that, unless P = NP, it is impossible to deduce in polynomial time a solution of the NDRSP from a vector $(x, y) \in I\!N^{|E|} \times \{0, 1\}^{|A|}$.

Corollary 3.2. *There is no 2-index formulation for the NDRSP.*

Proof. Let us suppose that there exists a 2-index formulation for the NDRSP. An optimal solution of this formulation would be a vector $(x, y) \in I\!N^{|E|} \times \{0, 1\}^{|A|}$. Since, from Theorem 3.1, a NDRSP solution can not be deduced in polynomial time from (x, y), then we reach a contradiction. □

4. Natural formulation and valid inequalities

The NDRSP can be formulated as a mixed-integer linear program where variables, called natural, are directly in correspondence with the edge of graph G: to each edge $e \in E$, we associate m binary variables x_e^t equal to 1 if e belongs to the ring-star t, 0 otherwise. To each assignment $(i, j) \in A$, we associate m binary variables y_{ij}^t equal to 1 if customer i is assigned to node j belonging to the ring-star t, 0 otherwise. We then consider the formulation (NF):

$$\text{Min} \sum_{t=1}^{m}\sum_{e\in E} l_e x_e^t + \sum_{t=1}^{m}\sum_{(i,j)\in A} c_{ij} y_{ij}^t$$

$$\sum_{t=1}^{m}\sum_{(i,j)\in A} y_{ij}^t = 1 \qquad \forall i \in U \tag{4.1}$$

$$\sum_{(i,j)\in A} d_i y_{ij}^t \leq Q \qquad \forall t = 1,\dots,m \tag{4.2}$$

$$\sum_{e\in\delta(i)} x_e^t \leq 2 \quad \forall i \in V,\ \forall t = 1,\dots,m, \tag{4.3}$$

$$\sum_{t=1}^{m} x_e^t \leq \gamma_e \qquad \forall e \in E \tag{4.4}$$

$$\sum_{e\in\delta(S)} x_e^t \geq 2 \sum_{(i,j)\in A, j\in S} y_{ij}^t \quad \forall i \in U,\ \forall t = 1,\dots,m \tag{4.5}$$

$$\forall S \subseteq V \setminus \{0\},\ S \neq \emptyset$$

$$x_e^t \in \{0,1\} \quad \forall e \in E,\ \forall t = 1,\dots,m,$$

$$y_{ij}^t \in \{0,1\}\ \forall (i,j) \in A,\ \forall t = 1,\dots,m,$$

Constraints (4.1) ensure that each customer is assigned to a single node, and a single ring-star. Constraints (4.2) ensure that the capacity of each ring-star is satisfied. Constraints (4.5) are the *connectivity inequalities*; they ensure that, for a given ring-star, there exist 2 edge-disjoint paths between the depot and any node with a client to which it has been assigned. Associated with degree constraints (4.3), constraints (4.5) ensure the ring structure.

The formulation (NF) is based on the Ring-Star problem formulation given by Labbe *et al.* in [12]. A polynomial time cutting plane based algorithm for solving the linear relaxation of (NF) will be presented in Section 6.

The linear relaxation of (NF) model can be strengthened through the introduction of valid inequalities.

- *1-Connectivity inequalities*

Figure 6 depicts a typical fractional solution obtained from the linear relaxation of (4.1)−(4.5). This solution can be seen as m identical subgraphs G^t where $G^t = (V^t, E^t, A^t), t = 1,\dots,m$ is given by the positive values of vectors x^t and y^t. Unfortunately subgraphs $(V(G^t), E(G^t))$, which are expected to be ring-stars, are only path-stars. Such fractional points are obtained when inequalities (4.5) are satisfied with equality.

This fractional solution structure leads to the following remark: given a non-empty subset $S \subseteq V \setminus \{0\}$, if an edge $e' \in \delta(S)$ is removed from the graph, there

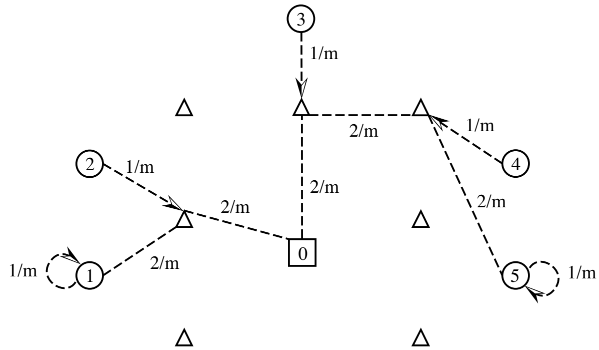


FIGURE 6. A fractional solution.

must exist a path from the depot to any node to which a client has been assigned. This remark shows that inequalities (4.6) are valid for NDRSP.

$$\sum_{e\in\delta(S)\setminus\{e'\}} x_e^t \geq \sum_{(i,j)\in A, j\in S} y_{ij}^t \quad \forall i\in U,\ \forall t=1,\dots,\ m, \qquad \forall S\subseteq V\setminus\{0\},\ S\neq\emptyset,\ \forall e'\in\delta(S) \tag{4.6}$$

Adding the following inequalities to the formulation permit to cut off the fractional point corresponding to the structure given in Figure 6. For instance, taking $S=\{1\}$ and e' as the edge in $\delta(1)$ with $x_{e'}^t = 2/m$ for an arbitrary t will provide a violated inequality (4.6).

- *Fractional capacity constraints*

In the well-known CVRP formulation given in [6], the main inequalities, called *capacity constraints*, limit the capacity requirement of the resulting CVRP tour. Among these inequalities, the fractionnal capacity contraints can be separated in polynomial time [2]. Since, in our case, the capacity of a tour depends on y variables values, these inequalities have to be adapted as follows:

$$\sum_{e\in\delta(S)} x_e^t \geq \frac{2}{Q}\sum_{(i,j)\in A, j\in S} d_i y_{ij}^t, \quad \forall t=1,\dots,\ m, \qquad \forall S\subseteq V\setminus\{0\},\ S\neq\emptyset \tag{4.7}$$

These inequalities ensure that a sufficient number of rings enter each subset of nodes and thus are valid for the NDRSP.

It is interesting to notice that inequalities (4.7) dominate the so-called CVRP fractional capacity constraints [2]. This can be easily proved by summing some of the inequalities (4.7) as follows. Let $\mathcal{U}\subsetneq U$ be a client subset. Let us consider a set $S\subsetneq V$ such that $\{j\in V : \exists(i,j)\in L_i, i\in\mathcal{U}\}\subset S$. By summing inequalities (4.7) corresponding to subset S for $t = 1, \ldots, m$, we then obtain

$$\sum_{t=1}^{m} \sum_{e \in \delta(S)} x_e^t \geq \frac{2d(\mathcal{U})}{Q}.$$

By letting $x_e = \sum_{t=1,\ldots,m} x_e^t, \forall e \in E$ the resulting inequality is exactly the CVRP fractional capacity constraint for subsets $\mathcal{U}$ and S.

- *Rounded capacity constraints*

In the CVRP formulation [6] the right hand side of the capacity constraint can be rounded up in order to obtain stronger inequalities. In the case of the NDRSP capacity constraint, such rounding operation cannot directly be done. In fact, for a given subset of clients $\mathcal{U} \subset U$, $\left\lceil \frac{d(\mathcal{U})}{Q} \right\rceil$ gives a lower bound on the number of rings required to satisfy the demand of these clients. Moreover, suppose that a quantity Q of the demand $d(\mathcal{U})$ is satisfied through a tour t', then the number of rings necessary to satisfy the remaining demand will be $\left\lceil \frac{d(\mathcal{U})}{Q} \right\rceil - 1$. This idea can be generalized as follows, removing a subset $\mathcal{T} \subsetneq \{1, \ldots, m\}$, the number of rings necessary to satisfy the remaining demand $d(\mathcal{U})$ will have to be greater than $\left\lceil \frac{d(\mathcal{U})}{Q} \right\rceil - |\mathcal{T}|$. Consequently, the following rounded capacity constraints are valid for NDRSP.

$$\sum_{t=1,\ t \notin \mathcal{T}}^{m} \sum_{e \in \delta(S)} x_e^t \geq 2\left(\left\lceil \frac{d(\mathcal{U})}{Q} \right\rceil - |\mathcal{T}|\right) \quad \forall \mathcal{U} \subsetneq U, \mathcal{U} \neq \emptyset, \tag{4.8}$$
$$\forall \mathcal{T} \subsetneq \{1, \ldots, m\},$$
$$\{j \in V : \exists (i,j) \in L_i, i \in \mathcal{U}\} \subset S.$$

By setting S as it has been done previously, and $\mathcal{T} = \emptyset$, we obtain the CVRP *rounded capacity constraints.*

5. Polyhedral study

In this section, we focus on the Non-Disjoint Ring-Star polytope which can be defined as the convex hull of the solutions of formulation (NF). Unfortunately, checking if NDRSP has a feasible solution is NP-complete, even if graph G is complete, consequently finding the dimension of this polytope is not easy. However we can study the polytope $\mathcal{P}$ defined as a relaxation of the fact that all clients must be served. Since the NDRSP polytope is a face of $\mathcal{P}$, studying $\mathcal{P}$ provides a useful structural insight of our valid inequalities.

$\mathcal{P}$ is then the convex hull of the solutions of formulation obtained from (NF) by replacing inequalities (4.1) by the following inequalities (5.1):

$$\sum_{t=1}^{m} \sum_{(i,j)\in A} y_{ij}^t \leq 1, \forall i \in U. \tag{5.1}$$

In the sequel, we will suppose that graph G is complete, with $n_u \geq 1$ and $n \geq 5$.

5.1. Basic properties

In this section we present basic properties for the facial description of the polytope.

We first introduce a short description of an integer point of $\mathcal{P}$. A NDRSP solution σ will be given by the positive variables: $\sigma = \left(\{x_e^t : x_e^t = 1\} \ ; \ \{y_{ij}^t : y_{ij}^t = 1\}\right)$.

Given an arc $(i, j) \in A$ and $t \in \{1, \ldots, m\}$, we define a particular solution χ_{ij}^t

$$\chi_{ij}^t = (x_{0j}^t, x_{jk}^t, x_{k0}^t \ ; \ y_{ij}^t),$$

where k is an arbitrary node. In fact, χ_{ij}^t is the incidence vector of a ring-star passing through the depot, node k, and node j to which client i is assigned. We note that such a solution will always exist under our assumptions.

Theorem 5.1. *$\mathcal{P}$ is full-dimensional.*

Proof. To show that $dim(\mathcal{P}) = m|E| + m|A|$ we have to exhibit $m|E| + m|A| + 1$ integer points of $\mathcal{P}$ whose incidence vectors are affinely independent.

First, remark that points $(x_e^t; \emptyset)$, for $t = 1, \ldots, m$, $e \in E$, are within $\mathcal{P}$. Moreover, for a given arc (i, j) and $t \in \{1, \ldots, m\}$, χ_{ij}^t is also clearly a point of $\mathcal{P}$. We then obtain $m|E| + m|A|$ integer points of $\mathcal{P}$ whose incidence vectors form the following matrix M where the columns corresponding to variables x are on the left, and those corresponding to variables y are on the right. Matrix M can be written

$$M = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ & & & 1 & \cdots & 0 \\ & M' & & 0 & \ddots & 0 \\ & & & 0 & \cdots & 1 \end{bmatrix}$$

where matrix M' is a 0/1 matrix. Since M is a nonsingular matrix, and $(\emptyset; \emptyset)$ is another point of $\mathcal{P}$, then $\mathcal{P}$ is full dimensional. $\square$

With similar arguments, we can easily obtain the following corollary.

Corollary 5.2. *Let* $t \in \{1, \dots, m\}$.

Given an edge $e \in E$*, the trivial inequality* $x_e^t \geq 0$ *is facet defining for* $\mathcal{P}$.

Given an arc $(i,j) \in A$*, the trivial inequality* $y_{ij}^t \geq 0$ *is facet defining for* $\mathcal{P}$.

We can now state that the degree inequalities are facet defining.

Theorem 5.3. *The degree inequalities (4.3) are facet defining for* $\mathcal{P}$.

Proof. Let $i_0 \in V$ and $t_0 \in \{1, \dots, m\}$. Let us denote by $ax + by \leq \alpha$ the degree inequality $\sum_{e \in \delta(i_0)} x_e^{t_0} \leq 2$. Suppose that there exists an inequality $a'x + b'y \leq \alpha'$ that is facet defining for $\mathcal{P}$ such that $\mathcal{F} = \{(x,y) \in \mathcal{P} : ax + by = \alpha\} \subset \mathcal{F}' = \{(x,y) \in \mathcal{P} : a'x + b'y = \alpha'\}$. We use Claim 5.4 to prove that the degree inequality $ax + by \leq \alpha$ is facet defining for $\mathcal{P}$.

Claim 5.4. Let $ax + by \leq \alpha$ (resp. $ax + by \geq \alpha$) be a valid inequality for $\mathcal{P}$. Suppose that there exists an inequality $a'x + b'y \leq \alpha'$ (resp. $a'x + b'y \geq \alpha'$) that is facet defining for $\mathcal{P}$ such that $\mathcal{F} = \{(x,y) \in \mathcal{P} : ax + by = \alpha\} \subset \mathcal{F}' = \{(x,y) \in \mathcal{P} : a'x + b'y = \alpha'\}$. Since, by Theorem 5.1, $\mathcal{P}$ is full dimensional, if there is $\lambda > 0$ such that $(a', b') = \lambda(a, b)$ and $\alpha' = \lambda\alpha$, then $ax + by \leq \alpha$ (resp. $ax + by \geq \alpha$) is facet defining for $\mathcal{P}$. $\Diamond$

Let $\{e_1, e_2\} \subset \delta(i_0)$ and $e_3 \in E \setminus \delta(i_0)$. Considering solutions $\sigma_1 = (x_{e_1}^{t_0}, x_{e_2}^{t_0}; \emptyset)$ and $\sigma_1' = (x_{e_1}^{t_0}, x_{e_2}^{t_0}, x_{e_3}^{t_0}; \emptyset)$, we then use Claim 5.5.

Claim 5.5. Let us consider two solutions σ and σ' such $ax^\sigma + by^\sigma = \alpha$ and $ax^{\sigma'} + bz^{\sigma'} = \alpha$. Then, by definition, both vectors also verify $a'x + b'y \leq \alpha'$ with equality and, consequently, $a'(x^\sigma - x^{\sigma'}) + b'(y^\sigma - y^{\sigma'}) = 0$. $\Diamond$

We obtain $a_{e_1}^{t_0} + a_{e_2}^{t_0} = a_{e_1}^{t_0} + a_{e_2}^{t_0} + a_{e_3}^{t_0}$ and then $a_e^{t_0} = 0, \forall e \in E \setminus \delta(i_0)$.

Let $e_4 \in E$ and $t' \neq t_0$. By considering the two solutions σ_1 and $\sigma_1'' = (x_{e_1}^{t_0}, x_{e_2}^{t_0}, x_{e_4}^{t'}; \emptyset)$, using Claim 5.5 we get that $a_e^{t'} = 0, \forall e \in E, t' \neq t_0$.

Let $(i,j) \in A$ and $t' \neq t_0$. By considering the two solutions σ_1 and $\sigma_2 = \sigma_1 + \chi_{ij}^{t'}$, using Claim 5.5 we get that $b_{ij}^{t'} = 0, \forall (i,j) \in A, t' \neq t_0$.

Let $(i', j') \in A$. Note that, since G is a complete graph, j' is either i_0 or $\{i_0, j'\} \in \delta(i)$. If $j' = i_0$, let $k \in V \setminus \{0, i_0\}$ and consider solutions $\sigma_2 = (x_{0i_0}^{t_0}, x_{i_0k}^{t_0}, x_{k0}^{t_0} \; ; \; y_{i'i_0}^{t_0})$, and $\sigma_2' = (x_{0i_0}^{t_0}, x_{i_0k}^{t_0}, x_{k0}^{t_0} \; ; \; \emptyset)$. If $j' \neq i_0$, consider solutions $\sigma_2 = (x_{0i_0}^{t_0}, x_{i_0j'}^{t_0}, x_{j'0}^{t_0} \; ; \; y_{i'j'}^{t_0})$, and $\sigma_2 = (x_{0i_0}^{t_0}, x_{i_0j'}^{t_0}, x_{j'0}^{t_0} \; ; \; \emptyset)$. In both cases, using Claim 5.5, we get that $b_{ij}^{t_0} = 0, \forall (i,j) \in A$.

Let $e_4 \in \delta(i)$. Considering the two solutions σ_1 and $\sigma_3 = (x_{e_1}^{t_0}, x_{e_4}^{t_0} \; ; \; \emptyset)$. Using Claim 5.5, we get that $a_{e_1}^{t_0} = a_{e_2}^{t_0}$. By letting $\lambda = a_{e1}^{t_0}$, we then get that $a_e^{t_0} = \lambda, \forall e \in \delta(i_0)$.

Finally, considering solution σ_1, we obtain $\alpha = 2\lambda$.

We have shown that $(a', b') = \lambda(a, b)$ and $\alpha' = \lambda\alpha$. Since the solution $(\emptyset, \emptyset)$ is valid for our polytope, $\lambda \geq 0$. Moreover, since $a'x + b'y \leq \alpha'$ is facet-defining, $\lambda > 0$. Then, using Claim 5.4, inequality $ax + by \leq \alpha$ is facet-defining for $\mathcal{P}$. $\square$

5.2. Connectivity inequalities

Unlike the previous inequalities, connectivity constraints (4.5) and 1-connectivity constraints (4.6) are not always facet defining. In order to obtain facet defining inequalities, we first give a technical definition.

• *2-cover set*
Given $S \subset V$, we define $U(S)$ as the client subset having an assignment arc ending in S, that is to say

$$U(S) = \{i \in U : \exists (i,j) \in L_i, j \in S\}.$$

A subset $C \subset U(S)$, will be called *2-cover set* if $d_i + d_j > Q$, $\forall i,j \in C,\ i \neq j$.
A 2-cover set C is *maximal* for $U(S)$ if for each j in $U(S) \setminus C$ there exists $i \in C$ such that $d_i + d_j \leq Q$. In order to give a short description of our inequalities, we introduce a last notation $A(C) = \{(i,j) \in A : i \in C, j \in S\}$, along with the following remark.

Remark 5.6. Let $S \subset V$ and $t \in \{1,\ldots,m\}$. For each 2-cover set $C \subset U(S)$, at most one client of C can be assigned to the ring-star t, that is to say

$$\sum_{(i,j)\in A(C)} y_{ij}^t \leq 1.$$

•*Generalized Connectivity inequalities*
Given $S \subset V \setminus \{0\}$ and $t \in \{1,\ldots,m\}$, the corresponding connectivity inequalities (4.5) can be strengthened by considering a 2-cover set $C \subset U(S)$ instead of a single client. In fact, from Remark 5.6, the generalized connectivity inequality

$$\sum_{e\in\delta(S)} x_e^t \geq 2 \sum_{(i,j)\in A(C)} y_{ij}^t \tag{5.2}$$

is valid for $\mathcal{P}$. Moreover, we have the following result.

Theorem 5.7. *The generalized connectivity inequalities (5.2) are facet defining for $\mathcal{P}$ if and only if (i) an (ii) hold:*

(i) C is maximal for $U(S)$;
(ii) if $|\bar{S}| < 3$, then for every $(i,j) \in A, j \in S$ or $i \in U \setminus C$.

Proof. Let $t_0 \in \{1,\ldots,m\}$, $S \subsetneq V, S \neq \emptyset$, and $C \subset U(S)$ a 2-cover set. If C is not maximal for $U(S)$, there exists a 2-cover set $C' \subset U(S)$ containing C such that the connectivity constraint corresponding to C' dominates the inequality corresponding to C. In the sequel we suppose that i) holds.

Let us now suppose that property (ii) does not hold, that is to say that $|\bar{S}| < 3$ and there exists $(i',j') \in A$ such that $j \in \bar{S}, i \in C$. Every valid solution such that

$y_{i'j'}^{t_0} = 1$ must use at least two edges $e, e' \in \delta(S)$. It follows that inequality (5.2) is clearly dominated by the following valid inequality:

$$\sum_{e \in \delta(S_0)} x_e^{t_0} \geq 2 \sum_{(i,j) \in A(C)} y_{ij}^{t_0} + 2y_{i'j'}^{t_0}.$$

Conversely, let us suppose that (i) and (ii) hold. Let us denote by $ax+by \geq 0$ the generalized connectivity constraint $\sum_{e\in\delta(S)} x_e^{t_0} \geq 2\sum_{(i,j)\in A(C)} y_{ij}^{t_0}$. We suppose that solutions (x, y) of $\{(x, y) \in P : ax + by = 0\}$ also satisfy $a'x + b'y = \alpha'$, where $a'x + b'y \geq \alpha'$ is facet defining for $\mathcal{P}$. Using Claim 5.4 we want to prove that $a'x + b'y \geq \alpha'$ is a multiple of $ax + by \geq 0$.

Since the solution $\sigma_1 = (\emptyset; \emptyset) \in \mathcal{F}$, we get that $\alpha' = 0$.

Let $e_1 \in E \backslash \delta(S)$, considering the solutions σ_1 and $\sigma_1' = (x_{e_1}^{t_0}; \emptyset)$, using Claim 5.5, we get that $a_e^{t_0} = 0, \forall e \in E \setminus \delta(S)$.

Let $e_2 \in E$ and $t' \neq t_0$, considering the solutions σ_1 and $\sigma_1'' = (x_{e_2}^{t'}; \emptyset)$, using Claim 5.5, we get that $a_e^{t'} = 0, \forall e \in E, t' \neq t_0$.

Let $(i, j) \in A$ and $t' \neq t_0$. Considering solutions σ_1 and $\chi_{ij}^{t'}$, using Claim 5.5, we get that $b_{ij}^{t'} = 0, \forall (i, j) \in A, t' \neq t_0$.

Let $(i, j) \in A(C)$ and $k, k' \in \bar{S} \setminus \{0\}$, $k \neq k'$. Since $|\bar{S}| \geq 3$, k and k' exist. Considering solutions $\sigma_2 = (x_{0j}^{t_0}, x_{jk}^{t_0}, x_{k0}^{t_0}; y_{ij}^{t_0})$ and $\sigma_2' = (x_{0j}^{t_0}, x_{jk'}^{t_0}, x_{k'0}^{t_0}; y_{ij}^{t_0})$, using Claim 5.5, we get that $a_{\{kj\}}^{t_0} = a_{\{k'j\}}^{t_0}$. By letting $\lambda = a_{\{kj\}}^{t_0}$, we then get that $a_e^{t_0} = \lambda, \forall e \in \delta(S)$.

Let $(i, j) \in A$ with $j \notin S$. Since $|\bar{S}| \geq 3$, there exists a node $k \neq j$, and we can consider solutions σ_1 and $\sigma_3 = (x_{0j}^{t_0}, x_{jk}^{t_0}, x_{k0}^{t_0}; y_{ij}^{t})$. Using Claim 5.5, we get that $b_{ij}^{t_0} = 0, \forall (i, j) \in A$ with $i \in U$ and $j \notin S$.

Let $(i, j) \in A$ with $j \in S$ and $i \notin C$. Since C is maximal for $U(S)$, $\exists (i', j') \in A(C)$ such that $d_i + d_{i'} \leq Q$. Let $k \in V \setminus \{0, j, j'\}$. We then consider solutions $\sigma_4 = (x_{0j'}^{t_0} + x_{j'j}^{t_0} + x_{j'k}^{t_0} + x_{k0}^{t_0}; y_{i'j'}^{t_0})$ and $\sigma_4' = (x_{0j'}^{t_0} + x_{j'j}^{t_0} + x_{j'k}^{t_0} + x_{k0}^{t_0}; y_{i'j'}^{t}, y_{ij}^{t})$. Using Claim 5.5, we get that $b_{ij}^{t_0} = 0, \forall i \in U \setminus C, j \in S$.

Finally, let $(i, j) \in A(C)$ and $(i', j') \in A(C)$, $(i, j) \neq (i', j')$ and $k \in V \setminus \{0, j, j'\}$. Considering solutions $\sigma_5 = (x_{0j'}^{t_0} + x_{j'j}^{t_0} + x_{jk}^{t_0} + x_{k0}^{t_0}; y_{ij}^{t_0})$ and $\sigma_5' = (x_{0j'}^{t_0} + x_{j'j}^{t_0} + x_{jk}^{t_0} + x_{k0}^{t_0}; y_{i'j'}^{t})$, using Claim 5.5, we get that $b_{ij}^{t_0} = b_{i'j'}^{t_0}, \forall (i, j), (i', j') \in A(C)$.

Considering solution σ_5, we also get that $b_{ij}^t = -2a_e^t, e \in \delta(S), \forall (i, j) \in A(C)$, that is to say $b_{ij}^t = -2\lambda, \forall (i, j) \in A(C)$.

We have shown that $(a', b') = \lambda(a, b)$. Moreover, let $e \in \delta(S)$, since the solution $(x_e^{t_0}, \emptyset)$ is valid for our polytope, we can deduce that $\lambda \geq 0$. Since $a'x + b'y \geq \alpha'$ is facet-defining, $\lambda > 0$. Then, using Claim 5.5, inequality $ax + by \geq \alpha$ is facet-defining for $\mathcal{P}$. $\square$

• *Generalized 1-Connectivity inequalities*

Using Remark 5.6, 1-connectivity constraints can be enforced so that the following inequality are valid:

$$\sum_{e\in\delta(S)\setminus\{e'\}} x_e^t \geq \sum_{i\in\mathcal{C}} \sum_{(i,j)\in A, j\in S} y_{ij}^t \ \forall S\subseteq V\setminus\{0\},\ S\neq\emptyset, \qquad (5.3)$$
$$\forall C\subset U(S),\ \forall e'\in\delta(S),\ \forall t=1,\ldots,\ m.$$

Using similar arguments as those used in the previous theorem proof, we then get a necessary and sufficient condition for inequality (5.3) to be facet defining.

Theorem 5.8. *The generalized 1-connectivity inequalities (5.3) are facet defining for $\mathcal{P}$ if and only if (i) an (ii) hold:*

(*i*) *C is maximal for U(S),*
(*ii*) *if $|\bar{S}| < 3$, then for every $(i,j)\in A, j\in S$ or $i\in U\setminus C$.*

6. Branch-and-Cut algorithm

In this section we present a Branch-and-Cut algorithm for the NDRSP. Our aim is to address the algorithmic application of the polyhedral results described in the previous sections.

The algorithm starts by solving a linear relaxation program containing assignment constraints (4.1), degree constraints (4.3) and trivial constraints. We then use separation algorithm for the generalized connectivity constraints (5.2). Usually the solution obtained at the end of this cutting plane phase is not integer, and thus, it is necessary to generate further additional inequalities: 1-connectivity (5.3) and capacity inequalities (4.7) or (4.8). If the solution is still fractional, we then use a Branch-and-Bound tree, where the additional inequalities (4.7), (4.8) and (5.3) are no longer separated. In what follows we present separation algorithms for the different classes of inequalities used by the algorithm as well as our branching strategy and primal heuristic.

• *Exact and heuristic separations*

Now we describe the separation routines used in the algorithm. These are exact algorithms except for the one used for the rounded capacity constraints (4.8). In what follows we denote $(\tilde{x},\tilde{y})$ an optimal (fractional) solution of the linear relaxation of (4.1)–(4.3).

We first have to give some notations. Given $t_0\in\{1,\ldots,m\}$ and a 2-cover set C, we denote by $\tilde{G}^{t_0}=(\tilde{V},\tilde{E},\tilde{A})$ the *support graph* associated with the partial solution $(\tilde{x}^{t_0},\tilde{y}^{t_0})$ defined as follows: we set $\tilde{V}$ as $V\cup\{n+1\}$, $\tilde{A}=\{(i,j)\in A : y_{ij}^{t_0}>0\}$, and $\tilde{E}=\{e\in E : x_e^{t_0}>0\}\cup E(C)$ where $E(C)=\{\{j,n+1\}, j\in\{j:(i,j)\in A(C)\}\}$.

From the following Lemma 6.1 it is sufficient to separate inequalities (5.2) and (5.3) only for subsets C which are maximal for U.

Lemma 6.1. *If $ax + by \leq 0$ is the connectivity constraint, for a ring-star $t_0 \in \{1, \ldots, m\}$, corresponding to subsets $S \subset V$ and a 2-cover set C maximal for $U(S)$, then there exists a 2-cover set C' maximal for U defining the same connectivity constraint, for the ring-star t_0, on subset $S \subset V$.*

The separation of the generalized connectivity constraint (5.2) can be performed in polynomial time. By extending the exact separation of connectivity constraints given in [12] for the Ring-Star problem, we can devise the following algorithm. Let C be a 2-cover set maximal for U, we denote by $\bar{Y}$ the following sum:

$$\bar{Y} = \sum_{t=1, t\neq t_0} \sum_{i\in C} \sum_{(i,j)\in A} y^t_{ij}.$$

Finding a most violated connectivity constraint $\sum_{e\in\delta(S_0)} x^{t_0}_e \geq 2\sum_{(i,j)\in A(C)} y^{t_0}_{ij}$ is equivalent to finding the largest violation of $\sum_{e\in\delta(S_0)} x^{t_0}_e + 2\bar{Y} \geq 2|C|$. This reduces to finding a maximum flow on graph $\tilde{G}^{t_0}$ where the capacity of each edge $e \in \tilde{E} \cap E$ is equal to $x^{t_0}_e$ and where the capacity of each edge $e = \{j, n+1\} \in E(C)$ is equal to $2\sum_{i\in C} y^{t_0}_{ij}$. Let $S' \subset \tilde{V} \setminus \{0\}$ be such that $n+1 \in S'$, and such that the capacity Δ of the cut $\delta(S')$ in $\tilde{G}^{t_0}$ is minimum. If $\Delta \geq 2|C|$, there is no connectivity constraint involving C violated by the current solution on the tour t_0. Otherwise, $S = S' \setminus \{n+1\}$ defines a most violated connectivity constraint (5.2) involving C. Consequently, by finding a maximum flow associated with each 2-cover set C maximal for U and with each $t \in \{1, \ldots, m\}$, we can exactly separate constraints (5.2) by solving at most $m \times n_u$ max flow problems.

Our separation algorithm for the generalized 1-connectivity constraints (5.3) uses ideas similar to the ones presented above. Consequently, by finding a maximum flow associated with each 2-cover set C maximal for U, with each edge $e \in E$ and with each $t \in \{1, \ldots, m\}$, we can exactly separate constraints (5.3) by solving at most $m \times n_u \times |E|$ maximum flow problems.

Now we turn our attention to the separation of capacity constraints (4.7) and (4.8). For the fractional capacity inequalities (4.7) an exact separation algorithm can be easily constructed using principles similar to the ones presented in [2] for the fractional capacity constraints of the CVRP. In the same way, a heuristic separation algorithm for inequalities (4.8) can be presented as follows: we first construct a procedure *create_ineq*$(\mathcal{U}, \mathcal{T})$, starting with a subset $\mathcal{U} \subset \{1, \ldots, n_u\}$ and subset $\mathcal{T} \subset \{1, \ldots, m\}$, that computes a set S such that $\tilde{x}(\delta(S)) = \sum_{e\in\delta(S)} \tilde{x}^t_e$ is minimum using a max flow algorithm. If $\tilde{x}(\delta(S)) < 2\left(\left\lceil \frac{d(\mathcal{U})}{Q} \right\rceil - |\mathcal{T}|\right)$ then a violated inequality (4.8) has been found. We use procedure *create_ineq*$(\mathcal{U}, \mathcal{T})$ for several values of $\mathcal{U}$ and $\mathcal{T}$ that are heuristically constructed.

- *Branching strategy*

The instances that we will address are dense and planar. In this type of instance, we have remarked that, when the y variables are integer, the x variables are also almost all integer at the end of the cutting-plane phase. That is related to the fact that if the assignments are known, our problem reduces to finding m Steiner TSP. In addition, the inequalities separated during the Branch-and-Cut phase are almost sufficient to give a partial characterization of the Steiner TSP polytope on these instances [3].

Consequently, we have chosen to branch first on the assignment variables y. We can further improve this branching rule by branching on assignment constraints:

$$\sum_{(i,j)\in L_i} y_{ij}^t = a, a \in \{0,1\}.$$

- *Primal heuristic*

We have also developed a primal heuristic in order to compute an integer solution, which corresponds to an upper bound for our problem. This heuristic works as follows.

For every $t = 1, \dots, m$ we first try to construct an elementary cycle $E(\mathcal{R}_t)$ with a greedy procedure which starts from the depot and constantly selects the edge having the highest value $\tilde{x}_e^t$ among the edges e incident to the last reached node. If this first step succeeds, we then try to determine, for every client $i \in U$, a pair (j', t') so that j' belongs to $E(\mathcal{R}_{t'})$ and $d(E(\mathcal{R}_{t'})) + d_i \leq Q$. In case there are several potential pairs (j', t'), we opt for the one with $y_{ij'}^{t'}$ of maximal value. Note that there is no guarantee that this heuristic produces a feasible solution.

7. Experimental results

The Branch-and-Cut algorithm described in the previous section was implemented in C++, using the SCIP 2.0 framework (see [1] for details on this software) using Cplex 12.1 as linear solver. The graph data structure was implemented using Lemon 1.2.1 library [7]. It was tested on PC Intel2 Duo running at 3.33 GHz with 8 Go RAM. We have considered two classes of instances, one randomly generated, while the other come from a SDH network corresponding to Paris and its surrounding business areas.

The first instances were generated in the following way:

Given the problem parameters $|U|$, $|W|$ and m, we generated vertices with coordinates in $[0, 100 * |V|] \times [0, 100 * |V|]$. We then consider a complete graph induced by this node set where the weight of each edge $\{i, j\}$ is computed as the Euclidean distance between nodes i and j. We sort the edges in decreasing order of weight and then delete every edge crossing a shorter one, thus the resulting graph is planar. We choose the assignments set $L_i = \{(i, j) : j \in \delta(i) \cup \{i\}\}$, where c_{ij} is the Euclidean distance between nodes i and j. Finally, we manually fix the capacity Q and values d_i such that there exist solutions using exactly m ring-stars. For each triplet $(m, |U|, |W|)$ we have generated twenty instances.

The realistic instances we have treated have been obtained from real data on a Parisian optical network belonging to one of the main European providers of wide band telecommunications. Since the real network is pretty large, we have

Table 1. Primal heuristic efficiency.

Instances			First solution				Best solution		
$\vert U\vert$	$\vert W\vert$	m	G1st	T1st	#N1st	H1st	TB^{st}	#NB^{st}	HB^{st}
10	10	3	7.48%	0.11'	9.55	70%	0.15'	40.55	60%
10	10	4	6.98%	0.15'	12.1	80%	0.35'	177	80%
10	10	5	6.69%	0.25'	18.15	95%	1.08'	683.35	75%
10	20	3	4.19%	0.33'	6.65	25%	0.35'	17.95	30%
10	20	4	5.75%	0.77'	19.25	90%	1.36'	221.4	95%
10	20	5	9.92%	1.31'	17.1	100%	3.51'	943.6	55%
10	40	3	3.92%	2.93'	7.15	35%	2.99'	16.7	20%
10	40	4	9.45%	4.66'	15.5	90%	5.56'	122.35	80%
10	40	5	5.70%	9.73'	23.45	100%	16.91'	763.35	75%
15	10	3	5.05%	0.5'	25.95	80%	0.74'	123.7	90%
15	10	4	14.74%	0.84'	22.7	95%	4.08'	867.65	95%
15	10	5	16.81%	1.55'	66.65	100%	51.19'	8703.65	100%
15	20	3	6.89%	1.64'	23.3	80%	2.89'	270.7	70%
15	20	4	15.66%	3.61'	25.3	100%	13.12'	1075.1	90%
15	20	5	15.22%	3.92'	82.6	100%	61.21'	5472.6	100%
15	40	3	8.08%	8.88'	24.7	85%	10.98'	185.75	75%
15	40	4	14.42%	14.37'	28.2	100%	34.03'	894.6	100%
15	40	5	14.39%	27.36'	79.7	100%	62.42'	1241.9	100%

Table 2. Cut efficiency.

Instances			SCIP without cuts				SCIP with cuts			
$\vert U\vert$	$\vert W\vert$	m	FG	RG	#N	TT	FG	RG	#N	TT
10	10	3	0%	22.59%	184.85	0.11'	0%	6.37%	74.05	0.16'
10	10	4	0%	29.52%	1820.2	1.07'	0%	7.22%	438.8	0.47'
10	10	5	0%	44.01%	9167.35	5.84'	0%	6.10%	1608.1	1.59'
15	10	3	0%	22.68%	419.15	0.68'	0%	7.91%	218.75	0.84'
15	10	4	0%	33.09%	15627.95	27.19'	0%	9.25%	2894.65	7.55'
15	10	5	16.53%	44.60%	33384.25	111.43'	4.80%	14.05%	21 733.7	98.44'

extracted only one area as follows. Let $\alpha \in]0,1]$ and a depot i, we have considered every network point (clients, interconnection points ...) whose distance from the depot is at most $\alpha \times l_{\max}$, where $l_{\max}$ is the maximum distance between two points of the instance. To ensure the feasibility of the resulting instance, it is necessary to add some edges until having a 2-connected graph. Using this procedure, we have obtained three realistic instances.

Tables 1–4 summarize the performance of our Branch-and-Cut algorithm on our instances. The first columns indicate the instance characteristics: the number of ring-stars m, clients $|U|$, and Steiner nodes $|W|$.

Table 1 focuses on the efficiency of our primal heuristic. It gives the relative gap (G1st) between the best known upper bound and the first feasible solution found

TABLE 3. Experimental results on randomly generated instances.

Instances				Gaps and solving time				Additional cuts			
$\|U\|$	$\|W\|$	m	O/P	FG	RG	#N	TT	# 1-con	# capa	# Rcapa	# con
10	10	3	20/20	0%	6.37%	74.05	0.16’	676.9	761.9	140	4324.5
10	10	4	20/20	0%	7.22%	438.8	0.47’	860.2	1009.2	128.1	15 308.5
10	10	5	20/20	0%	6.10%	1608.1	1.59’	1298.2	1487.1	133.3	40 082.5
10	20	3	20/20	0%	4.71%	37.35	0.37’	900.8	1218.9	197.3	4166.1
10	20	4	20/20	0%	7.69%	687.45	1.99’	1594.2	2150.0	241.3	46 145.1
10	20	5	20/20	0%	6.21%	1710.2	4.37	1967.9	2649.0	233.7	66 405.0
10	40	3	20/20	0%	4.29%	37.25	3.05’	1966.9	2994.6	439.9	9369.4
10	40	4	20/20	0%	7.24%	309.8	6.31’	2719.3	4278.9	417.1	46 958.7
10	40	5	20/20	0%	5.68%	1705.15	21.24’	3696.7	5690.3	442.1	174 124.5
15	10	3	20/20	0%	7.91%	218.75	0.84’	1083.3	757.0	211.4	16 444.4
15	10	4	20/20	0%	9.25%	2894.65	7.55’	1584.4	1337.8	227.1	178 804.9
15	10	5	5/20	0%	14.05%	21733.7	98.44’	2123.7	1763.5	236.5	2 296 017.6
15	20	3	20/20	0%	8.92%	429.25	3.16’	1626.0	1286.2	322.8	535 404.1
15	20	4	20/20	0%	9.28%	5077.9	29.05’	2293.7	2310.6	346.6	506 289.7
15	20	5	3/20	6.63%	14.41%	11904	110.05’	2738.4	2744.8	331.9	2 162 005.9
15	40	3	20/20	0%	8.06%	319.6	11.69’	2777.6	2515.2	600.5	75 123.4
15	40	4	16/20	0.70%	14.42%	2670.75	54.6’	3669.0	4257.8	580.1	593 106.1
15	40	5	1/20	13.25%	14.39%	3389.3	119.58’	5640.3	6545.0	645.7	1 289 893.4

TABLE 4. Experimental results on realistic instances.

Instances				Gaps and solving time				Additional cuts			
α	$\|U\|$	$\|W\|$	m	FG	RG	#N	TT	# 1-con	# capa	# Rcapa	# con
5%	23	104	3	0%	12.01%	1903	10.60’	2789	3118	1616	226 836
5.3%	35	127	3	0%	7.61%	5442	64.73’	9503	9096	3782	1 018 048
5.4%	42	136	3	18.33%	21.98%	7300	120’	7668	7527	4914	2 473 000

during the Branch-and-Cut process. The other columns give more details about those two solutions:

T1^{st} (*resp.* $\text{T}B^{st}$)	:	the total CPU time in minutes needed to get the first (*resp.* best) solution.
$\#\text{N1}^{st}$ (*resp.* $\#\text{N}B^{st}$)	:	the number of nodes of the Branch-and-Bound tree treated before getting the first (*resp.* best) solution.
H1^{st} (*resp.* $\text{H}B^{st}$)	:	the number of instances for which our heuristic found the first (*resp.* best) solution.

It can be deduced from Table 1 that for 77% (*resp.* 85%) of the instances, the best (*resp.* first) solution has been provided by our primal heuristic. Note that, since there is no guarantee that our heuristic produces a feasible solution, the first solution may not be computed at the root node and may be determined by one of the SCIP heuristics. The average relative gap between the first and the best solution is 9.5%, and the highest gap is 16.78%. The first solution is obtained in less than thirty minutes. Note that, in most cases, this first solution is in fact obtained in less than 10 minutes. This proves that our algorithm is able to quickly provide a good solution. Furthermore, the efficiency of this heuristic permits to limit the size of the Branch-and-Bound tree.

For the remaining tables, the columns give the following results:

O/P	:	the number of problems solved to optimality over the number of instances tested,
FG	:	the relative gap between the best known upper and lower bounds,
RG	:	the relative gap between the best known upper bound and the lower bound achieved before branching,
#N	:	the number of nodes of the Branch-and-Bound tree,
TT	:	the total CPU time in minutes within a time limit of two hours,
#con	:	the number of connectivity constraints separated,
#ineq	:	the number of constraints of each family separated at the root node of the Branch-and-Bound tree.

The two parts of Table 2 indicate the results obtained with and without using our additional 1-connectivity cuts (5.3) and capacity cuts (4.7), (4.8). This table focuses on lines $(m, |U|, |W|)$ where at least one instance among the twenty instances is both solved to optimality with or without our additional inequalities. We can notice that the inequalities we introduced really improve both the lower bound obtained from the linear relaxation and the size of the Branch-and-Bound tree. Moreover the total CPU time is globally better or similar. This shows that the separation procedures we developed for these inequalities are effective.

Table 3 presents the results on the randomly generated instances. Each line gives the average values obtained from twenty instances. For each size, at least one instance was solved to optimality within the time limit and for all of them a valid solution has been provided. We have solved instances up to 15 clients, 40 Steiner nodes and 5 ring-stars. We can remark that the size of the Branch-and-Bound tree strongly depends on the value of m. This follows from the fact that the symmetry of our formulation inscreases as a function of m. Note that the number of connectivity constraints that have to be generated during the whole Branch-and-Bound phase reaches more than 2 millions.

Finally, Table 4 gives the results obtained for realistic instances. We can first remark that they seem to be easier than the ones randomly generated. Our Branch-and-Cut algorithm succeeds in solving network instances corresponding to a small town area (5% of the big real network).

8. Concluding remarks

In this article we have studied the Non-Disjoint (Steiner) m-Ring-Star Problem. We have shown that this problem is the combinatorial structure of SDH networks which can be seen as the practical implementation of reliability for telecommunication networks. We have discussed about the possible integer formulations and shown that they cannot be written with two index variables.

We have given a natural integer formulation for the problem and propose additional inequalities. Some polyhedral results have been established about a full dimensional polytope containing the NDRSP polytope as a face. In particular we have given a generalization of the basic connectivity inequalities together with necessary and sufficient conditions for this generalized inequalities to be facet defining. We have constructed a Branch-and-Cut algorithm which consists in several efficient exact separation algorithms. A dedicated branching strategy along with an efficient primal heuristic were also provided. Our experimental results have shown that our Branch-and-Cut algorithm succeeds in solving network instances corresponding to realistic networks. However, this method must be improved in order to solve instances on bigger graphs.

Since the NDRSP cannot be solved with a two index variables and suffers from symmetry with an increase of the number of ring-stars, it will be interesting to try another approach like a column generation based algorithm, especially for instances when the number of ring-stars is greater than 4.

References

[1] T. Achterberg, SCIP: solving constraint integer programs. *Math. Program. Comput.* **1** (2009) 1–41.

[2] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán and D. Naddef, Separating capacity constraints in the CVRP using tabu search. *Eur. J. Oper. Res.* **106** (1998) 546–557.

[3] M. Baïou and A.R. Mahjoub, The Steiner traveling salesman polytope and related polyhedra. *SIAM J. Opt.* **13** (2002) 498.

[4] R. Baldacci, M. Dell'Amico and J.S. Gonzalez, The capacitated m-ring-star problem. *Oper. Res.* **55** (2007) 1147.

[5] R. Baldacci, E. Hadjiconstantinou and A. Mingozzi, An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Oper. Res.* **52** (2004) 723–738.

[6] G. Cornuejols and F. Harche, Polyhedral study of the capacitated vehicle routing problem, on the p-median polytope. *Math. Program.* **60** (1991) 21–52.

[7] B. Dezső, A. Jüttner and P. Kovács, Lemon-an open source c++ graph template library. *Electronic Notes in Theoretical Comput. Sci.* **264** (2011) 23–45.

[8] R. Fukasawa, H. Longo, J. Lysgaard, M.P. Aragão, M. Reis, E. Uchoa and R.F. Werneck, Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. program.* **106** (2006) 491–511.

[9] M.R. Garey and D.S. Johnson, *Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences*. WH Freeman and Company, San Francisco, Ca (1979).

[10] E.A. Hoshino and C.C. de Souza, A branch-and-cut-and-price approach for the capacitated m-ring-star problem. *Discrete Appl. Math.* **160** (2012) 2728–2741.

[11] S. Kedad-Sidhoum and V.H. Nguyen, An exact algorithm for solving the ring star problem. *Optimization* **59** (2010) 125–140.

[12] M. Labbé, G. Laporte, I.R. Martin and J.J.S. González, The ring star problem: Polyhedral analysis and exact algorithm. *Networks* **43** (2004) 177–189.

[13] G. Laporte, The vehicle routing problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **59** (1992) 345–358.

[14] A.N. Letchford and J.J. Salazar-Gonzalez. Projection results for vehicle routing. *Math. Program.* **105** (2006) 251–274.

[15] V. Hung Nguyen and M. Minoux, New formulation for the sonet/sdh network design problem, in *Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision* (2006).

[16] P. Soriano, C. Wynants, A. Seguin *et al.* Design and dimensionning of survivable SDH/SONET networks, in *Telecommunications network planning* (1999), pp. 147–167.