The Ring/κ-Rings Network Design Problem: Model and Branch-and-Cut Algorithm

Inmaculada Rodríguez-Martín and Juan-José Salazar-González
DMEIO, Facultad de Ciencias, Universidad de La Laguna, Tenerife, Spain

Hande Yaman
Department of Industrial Engineering, Bilkent University, Ankara, Turkey

This article considers the problem of designing a two-level network where the upper level consists of a backbone ring network connecting the so-called hub nodes, and the lower level is formed by access ring networks that connect the non-hub nodes to the hub nodes. There is a fixed cost for each type of link, and a facility opening cost associated to each hub. The number of nodes in each access ring is bounded, and the number of access rings connected to a hub is limited to κ, thus resulting in a ring/ κ-rings topology. The aim is to decide the hubs to open and to design the backbone and access rings to minimize the installation cost. We propose a mathematical model, give valid inequalities, and describe a branch-and-cut algorithm to solve the problem. Computational results show the algorithm is able to find optimal solutions on instances involving up to 40 nodes within a reasonable time. **NETWORKS, Vol. 68(2), 130–140 2016**

Keywords: network design; ring networks; valid inequalities; branch-and-cut

1. INTRODUCTION

This article shows the results of our research on the problem of designing a two-level network in which both the upper level network, that connects the hubs, and the lower level networks that connect user nodes to hubs, are rings. Rings are common structures in telecommunication networks as they provide survivability against single edge/node failures. Synchronous digital hierarchy (SDH), synchronous optical network (SONET), and wavelength-division multiplexing systems and Resilient Packet Rings are often configured in self-healing rings. Large networks use hierarchy to connect rings at different levels. Rings are also common in transportation applications where vehicles start and end their trips at a depot. Multi-echelon distribution systems, as in multimodal networks and city logistics, involve the design of rings at different levels of a transportation network where intermediate facilities are used for storage and transfer activities (see, e.g., [6]). In our study, we use the telecommunications terminology and refer to the upper level network as *backbone* and to the lower level networks as *access networks*.

In our problem, we are given an undirected graph, costs of installing hubs at nodes, and costs of installing both backbone and access edges. The problem requires selecting a subset of nodes as hubs, connecting the hubs with a ring network, and connecting the remaining nodes to hubs with rings. We impose that an access ring may contain at most q nodes (including its hub). At least one and at most κ access rings can be adjacent to a hub. The aim is to minimize the total cost of installing hubs and backbone and access edges.

Klincewicz [17] uses the notation "backbone structure/access structure" to specify the structure of a two-level network. For instance, in a "fully connected/ring network," the backbone network is a complete graph on the hubs, and the access networks are rings, each visiting a subset of users and one hub. With this notation, a solution to our problem is called a "ring/ κ-rings" network, and the problem is named *ring/ κ-rings network design problem*. Figure 1 shows a ring/3-rings optimal solution network for an instance with 20 nodes where the number of nodes in each access network is limited to 5. The solid lines represent the backbone network and the dashed lines represent the access networks. The nodes in the backbone network are the hubs. In this solution, one of the hubs has three access networks (the maximum allowed), while the other hubs have only one access network. Figure 2 shows the optimal solution for the same instance when the required network structure is ring/2-ring, that is, when at most two access ring per hub are permitted. The optimal solution of the ring/1-ring problem is depicted in Figure 3.

Even though hierarchical networks are common in many applications, there are few studies on related problems where

Received October 2015; accepted May 2016
Correspondence to: I. Rodríguez-Martín; e-mail: irguez@ull.es
Contract grant sponsor: "Ministerio de Economía y Competitividad", Spain;
Contract grant number: MTM2012-36163-C06-01
Contract grant sponsor: Turkish Academy of Sciences (to H.Y.)
DOI 10.1002/net.21687
Published online 28 June 2016 in Wiley Online Library (wileyonlinelibrary.com).

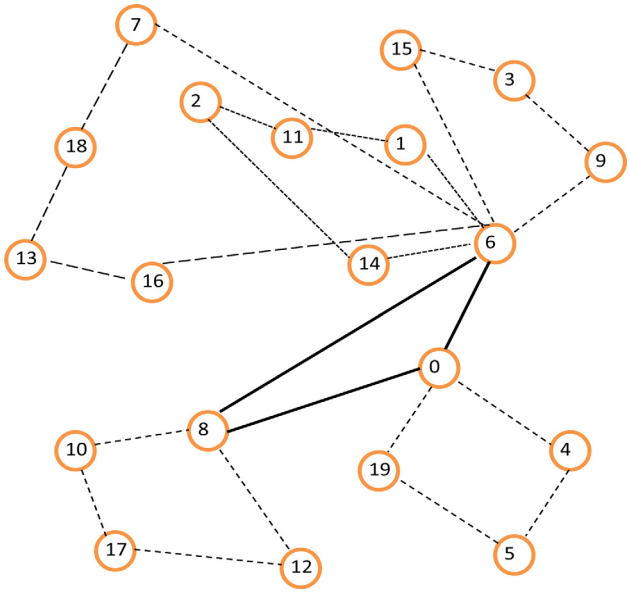

FIG. 1. A ring/3-rings solution example (cost: 1,501). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

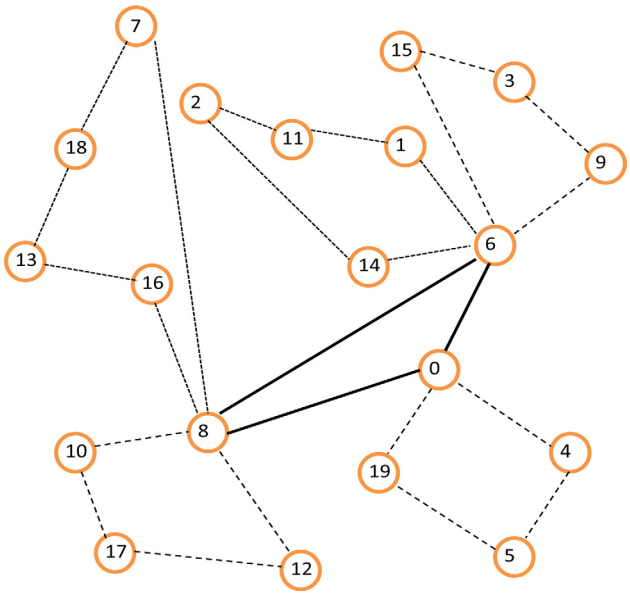

FIG. 2. A ring/2-rings solution example (cost: 1,502). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the networks are rings in both levels. Most of the studies that consider survivability are on the design of a single layer network (see, e.g., [13, 15, 16, 22, 23, 29, 31]). Fortz and Labbé [7] and Fortz et al. [8–10] study the problems of designing rings of bounded size. Labbé et al. [18] consider the problem of simultaneously locating plants and designing rings that connect customers to those plants. The number of customers that a plant can serve is limited. Magnanti and Raghavan [21] and Balakrishnan et al. [1, 2] consider more general survivability requirements. Klincewicz [17] reviews combined hub location and network design problems. Gourdin et al. [12] survey the studies on location problems encountered in

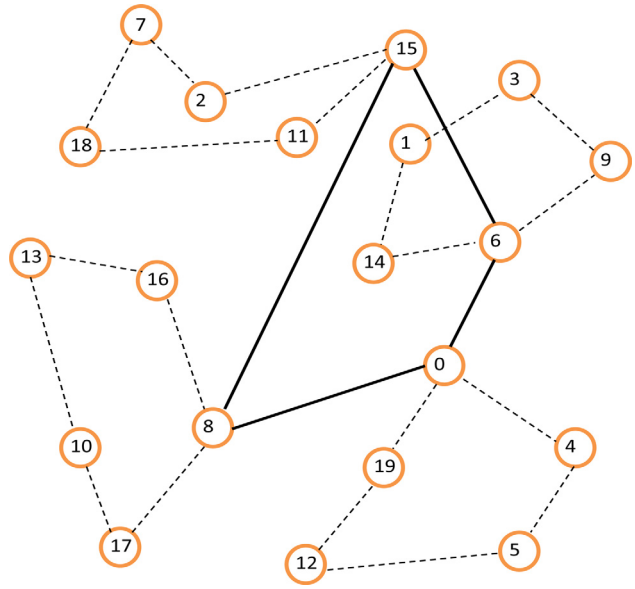

FIG. 3. A ring/1-ring solution example (cost: 1,713). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

telecommunications network design. Soriano et al. [28] provide an overview of design and dimensioning problems in survivable SDH/SONET networks. Contreras and Fernández [5] review studies on combined location and network design problems.

Labbé et al. [19] study the design of a ring/star network and propose a branch-and-cut algorithm. Baldacci et al. [3] study a generalization where the backbone network allows m rings instead of one. Lee and Koh [20] study the ring/chain network design problem with dual homing for a given ring backbone. They show that the problem related with the design of access networks is NP-hard, give a formulation and a tabu search algorithm. Thomadsen and Stidsen [30] study the ring/1-ring network design problem with an objective function that includes the cost for installing edges and also savings obtained by handling communication demands within access rings. Due to the complexity introduced by the savings in the objective function, they propose to solve the problem in two steps. They describe a branch-and-price algorithm for the optimal design of the access rings in the first step. The design of the backbone ring is modeled as a Generalized Traveling Salesman Problem in the second step. While the first step is deeply investigated in [30], the second step is omitted. Our approach to the ring/1-ring network design problem tackles the two steps simultaneously, as an integrated problem, although without considering savings in the objective function. Rodríguez-Martín et al. [25] propose a branch-and-cut algorithm for several two-level network design problems with survivability requirements in both levels. Two of these problems are ring/two-edge connected network design and ring/ring network design problems. However, the ring/ring problem in [25] differs from the ring/1-ring problem studied here (i.e., the ring/ κ-rings where $\kappa = 1$) since in the

former not all nodes must be necessarily on an access network, giving rise to optimal solutions with hubs without any adjacent access network. Carroll and McGarraghy [4] propose to decompose the problems of designing the rings in different levels. Shi and Fonseka [26] study the design of hierarchical self healing rings and propose a heuristic. Proestki and Sinclair [24] and Shi and Fonseka [27] propose heuristic algorithms for the problem with dual homing. Fouilhoux et al. [11] and Karasan et al. [14] study the design of two level networks with single and dual homing and a survivable backbone network.

In this article, we generalize the ring/ring network design problem by allowing multiple rings at each hub. We present a formulation and valid inequalities to strengthen it. We propose a branch-and-cut algorithm to solve this problem and analyze the results of our computational experiments.

The remainder of the article is organized as follows. We present our formulation in Section 2. In Section 3, we derive several families of valid inequalities to strengthen the formulation. We explain the details of our branch-and-cut algorithm in Section 4, and present our computational results in Section 5. Finally, the article ends with conclusions in Section 6.

2. FORMULATION

We first define the problem formally and then present the formulation. Let $V = \{0, 1, \ldots, n-1\}$ be the set of nodes. Node 0 is the root node and a hub is already installed at this node. Let $E = \{\{i,j\} : i,j \in V, i < j\}$ be the set of edges. We assume that all edges are potential links for a solution. Let $G = (V,E)$ be an undirected graph with no multiple edges allowed. We denote with f_j the fixed cost of installing a hub at node $j \in V$, and with b_e and a_e the costs of installing a backbone link and an access link on edge $e \in E$, respectively.

The aim of our problem is to find a subset of nodes of V on which hubs are installed, to connect these hubs with backbone edges that form a ring, to assign each non-hub node to a hub node, and to connect the nodes that are assigned to the same hub with access rings. Each access ring must contain at most q nodes, and each hub must be adjacent to at least one and to at most κ access rings. We would like to choose the hubs and the backbone and access rings in such a way that the total design cost is minimized.

Let $K = \{1, \ldots, \kappa\}$. The decision variables in our formulation are the following. The variable z_{ij} takes value 1 if node $i \in V$ is assigned to hub $j \in V$, and 0 otherwise. If $z_{jj} = 1$ then node j is a hub itself. For $i \neq j$ and $k \in K$, w_{ij}^k is equal to 1 if node i is assigned to the k th ring adjacent to hub node j, and 0 otherwise. For $i \in V$, w_{ii}^k is 1 if the k th ring adjacent to node i is used, and 0 otherwise. With these definitions, we have that $z_{ij} = \sum_{k \in K} w_{ij}^k$ for all $i,j \in V$, $i \overline{=} j$, and $z_{jj} = \max_{k \in K} w_{jj}^k$ for all $j \in V$. Note that the latter enforces at least one access ring for each hub. Regarding the edge design variables, we define x_e to be 1 if edge $e \in E$ is used in an access network and 0 otherwise, and y_e to be 1 if edge e is used in the backbone network and 0 otherwise.

We also use the following notation. For $S \subseteq V$, $\delta(S)$ is the set of edges with one endpoint in S, and $E(S)$ is the set of edges with both endpoints in set S. When S is a singleton, that is, $S = \{i\}$, we use $\delta(i)$ instead of $\delta(\{i\})$. For brevity, we write $x(E') = \sum_{e \in E'} x_e$ and $y(E') = \sum_{e \in E'} y_e$ for all $E' \subseteq E$, and $z(S:T)$ instead of $\sum_{i \in S, j \in T} z_{ij}$ for all $S, T \subseteq V$.

The ring/ κ-ring network design problem can be modeled as follows:

$$\min \sum_{i \in V} f_i z_{ii} + \sum_{e \in E} a_e x_e + \sum_{e \in E} b_e y_e \tag{1}$$

$$\text{s.t. } z(i:V) = 1 \quad \forall i \in V, \tag{2}$$

$$\sum_{i \in V} w_{ij}^k \leq q\, w_{jj}^k \quad \forall j \in V, k \in K, \tag{3}$$

$$z_{ij} = \sum_{k \in K} w_{ij}^k \quad \forall i \in V, j \in V \setminus \{i\}, \tag{4}$$

$$z_{jj} = w_{jj}^1 \geq \cdots \geq w_{jj}^\kappa \quad j \in V, \tag{5}$$

$$z_{00} = 1, \tag{6}$$

$$x(\delta(i)) = 2 \sum_{k \in K} w_{ii}^k + 2z(i : V \setminus \{i\}) \quad \forall i \in V, \tag{7}$$

$$y(\delta(i)) = 2z_{ii} \quad \forall i \in V, \tag{8}$$

$$z_{ij} + y_{\{i,j\}} \leq z_{jj} \quad \forall i,j \in V : \{i,j\} \in E, \tag{9}$$

$$z_{ji} + y_{\{i,j\}} \leq z_{ii} \quad \forall i,j \in V : \{i,j\} \in E, \tag{10}$$

$$y(\delta(S)) \geq 2z(i:S) \quad \forall S \subseteq V \setminus \{0\}, i \in S, \tag{11}$$

$$x(\delta(S)) \geq 2z(i : V \setminus S) \quad \forall S \subset V, i \in S, \tag{12}$$

$$x_{\{i,j\}} + w_{ii'}^k + w_{jj'}^{k'} \leq 2 \quad \forall \{i,j\} \in E, i', j' \in V \text{ and } k, k' \in K : (i' \neq j') \text{ or } (i' = j', i' \neq i, j' \neq j, k \neq k'), \tag{13}$$

$$x_{\{i,j\}} + z_{ii} + z(j : V \setminus \{i\}) \leq 2 \quad \forall \{i,j\} \in E, \tag{14}$$

$$x_{\{i,j\}} + z(i : V \setminus \{j\}) + z_{jj} \leq 2 \quad \forall \{i,j\} \in E, \tag{15}$$

$$w_{ij}^k \in \{0,1\} \quad \forall i,j \in V, k \in K, \tag{16}$$

$$x_e, y_e \in \{0,1\} \quad \forall e \in E, \tag{17}$$

$$z_{ij} \in \{0,1\} \quad \forall i,j \in V. \tag{18}$$

The objective function (1) is the sum of the cost of locating hubs and the cost of installing access and backbone edges. Constraints (2) ensure that each node is either a hub or it is assigned to another hub node. Constraints (3) are capacity constraints that limit the number of nodes on each access ring to q. They also ensure that no node is assigned to a non existing ring. Constraints (4) and (5) relate the variables z and w. If a node is assigned to a hub different than itself, then it is assigned to exactly one of the rings adjacent to that hub. If there is a ring adjacent to a node, then the node is a hub. Note that constraints (5) force the assignment of consecutive labels (1 to κ) to the rings adjacent to a hub, thus breaking some symmetries. They also make sure that each hub has at least one access ring adjacent to it. Constraint (6) forces the

root node to be a hub. Constraints (7) and (8) are degree constraints. The number of access edges adjacent to a node is two if the node is assigned to another hub. If the node is a hub itself, then it is twice the number of rings adjacent to it. The number of backbone links adjacent to a node is two if the node is a hub and it is zero otherwise. If an edge is used in the backbone network then constraints (9) and (10) ensure that both endpoints should be hubs. Otherwise, one endpoint can be assigned to the other only if the latter is a hub. Constraints (11) impose the 2-edge connectedness of the backbone network. If node $i \in S$ is a hub or if it is assigned to a hub node in set S, then there exists at least one hub in set S and the constraint asks for at least two backbone edges on the cut $\delta(S)$ since the root is in $V \setminus S$. Similarly, constraints (12) ensure the 2-edge connectedness of the access networks. If node $i \in S$ is allocated to a hub node in $V \setminus S$ then there should be at least two access links between S and $V \setminus S$. By imposing 2-edge connectedness together with the degree constraints, we make sure that the networks are rings. Constraints (13)–(15) ensure that if the access link $\{i,j\}$ is used then i and j are allocated to the same ring adjacent to the same hub. Finally, constraints (16)–(18) are variable restrictions.

3. VALID INEQUALITIES

Let $\mathcal{X}$ be the feasible set of model (2)–(18). In this section, we propose several families of valid inequalities for $\mathcal{X}$ that generalize other inequalities given in Rodríguez-Martín et al. [25] for the ring/ring network design problem.

The valid inequalities of the first family guarantee that an access edge cannot be used if its endpoints are in two different rings.

Theorem 1. *Let $\{i,j\} \in E$ and $K_l \subseteq K$ for all $l \in V \setminus \{i,j\}$. The inequality*

$$x_{\{i,j\}} \leq \sum_{l \in V \setminus \{i,j\}} \left(\sum_{k \in K_l} w_{il}^k + \sum_{k \in K \setminus K_l} w_{jl}^k \right) + z_{ij} + z_{ji} \qquad (19)$$

is valid for $\mathcal{X}$. In addition, these inequalities dominate constraints (13), and constraints (14) and (15) are special cases.

Proof. Suppose that $\sum_{l \in V \setminus \{i,j\}} (\sum_{k \in K_l} w_{il}^k + \sum_{k \in K \setminus K_l} w_{jl}^k) + z_{ij} + z_{ji} = 0$. We prove that $x_{\{i,j\}} = 0$ in the three possible cases:

1. $z_{ii} = 1$. In this case, as $z_{ji} = 0$, we know that j is not assigned to hub i and so we cannot use edge $\{i,j\}$.
2. $z_{jj} = 1$. Similar to the previous case.
3. $z_{ii} = z_{jj} = 0$. Then, there exists $l \in V \setminus \{i,j\}$ such that $\sum_{k \in K \setminus K_l} w_{il}^k = 1$ and $l' \in V \setminus \{i,j\}$ such that $\sum_{k \in K_{l'}} w_{jl'}^k = 1$. If l and l' are different, then nodes i and j are assigned to different hubs and edge $\{i,j\}$ cannot be used. If l and l' are the same, then i and j are assigned to two different rings adjacent to node l and again we cannot use edge $\{i,j\}$.

This proves the validity of the inequality.

Next, we prove that inequality (19) is stronger than (13). To see this, we first rewrite (13) for $i', j' \in V$, $k, k' \in K$ such that $i' \neq j'$, or $i' = j'$, $i' \neq i$, $j' \neq j$, and $k \neq k'$, as

$$x_{\{i,j\}} \leq z(i : V \setminus \{i'\}) + \sum_{\hat{k} \in K \setminus \{k\}} w_{ii'}^{\hat{k}} + z(j : V \setminus \{j'\}) + \sum_{\hat{k} \in K \setminus \{k'\}} w_{jj'}^{\hat{k}}. \qquad (20)$$

Now, consider inequality (19) for $K_{i'} = K \setminus \{k\}$, $K_{j'} = \{k'\}$ and $K_l = K$ for $l \in V \setminus \{i,j,i'j'\}$. We obtain

$$x_{\{i,j\}} \leq z(i : V \setminus \{i,j,i',j'\}) + \sum_{\hat{k} \in K \setminus \{k\}} w_{ii'}^{\hat{k}} + w_{ji'}^k + w_{ij'}^{k'} + \sum_{\hat{k} \in K \setminus \{k'\}} w_{jj'}^{\hat{k}} + z_{ij} + z_{ji}. \qquad (21)$$

Now as

$$z(i : V \setminus \{i,j,i',j'\}) \leq z(i : V \setminus \{i',j',j\}),$$

$$w_{ji'}^k + \sum_{\hat{k} \in K \setminus \{k'\}} w_{jj'}^{\hat{k}} + z_{ji} \leq z(j : V \setminus \{j'\}) + \sum_{\hat{k} \in K \setminus \{k'\}} w_{jj'}^{\hat{k}},$$

$$\sum_{\hat{k} \in K \setminus \{k\}} w_{ii'}^{\hat{k}} + w_{ij'}^{k'} + z_{ij} \leq \sum_{\hat{k} \in K \setminus \{k\}} w_{ii'}^{\hat{k}} + z_{ij'} + z_{ij},$$

the right-hand side of (21) is not more than the right-hand side of (20). Hence inequalities (19) dominate constraints (13).

Finally, we prove that constraints (14) and (15) are special cases of inequalities (19). Let $\{i,j\} \in E$. Using constraints (2) for nodes i and j, constraint (14) can be rewritten as

$$x_{\{i,j\}} \leq z(i : V \setminus \{i,j\}) + z_{ij} + z_{ji}.$$

This is inequality (19) for $K_l = K$ for all $l \in V \setminus \{i,j\}$. The proof for constraints (15) can be done similarly. ∎

The second family of valid inequalities strengthens the connectivity constraints (12) for the access networks.

Theorem 2. *Let $S \subset V$ be a non-empty set. Let $(S_1, \ldots, S_{m_1})$ be a partition of $S \times K$ and $(T_1, \ldots, T_{m_2})$ be a partition of $(V \setminus S) \times K$. Consider $i_1, \ldots, i_{m_2}$ distinct nodes in S and $j_1, \ldots, j_{m_1}$ distinct nodes in $V \setminus S$. The inequality*

$$x(\delta(S)) \geq 2 \left(\sum_{l=1}^{m_2} \sum_{(u,k) \in T_l} w_{i_l u}^k + \sum_{l=1}^{m_1} \sum_{(u,k) \in S_l} w_{j_l u}^k \right) \qquad (22)$$

is valid for $\mathcal{X}$. Inequality (22) with $m_2 = 1$ and $i_1 = i$ dominates constraint (12) for any i in S.

Proof. As nodes $i_1, \ldots, i_{m_2}$ and $j_1, \ldots, j_{m_1}$ are all distinct and sets $S_1, \ldots, S_{m_1}$ and $T_1, \ldots, T_{m_2}$ are all disjoint, $\sum_{l=1}^{m_2} \sum_{(u,k) \in T_l} w_{i_l u}^k + \sum_{l=1}^{m_1} \sum_{(u,k) \in S_l} w_{j_l u}^k$ is a lower bound on the number of rings that cross the cut $\delta(S)$. As each ring

uses at least two edges in $\delta(S)$, the inequality is satisfied by all feasible solutions.

If we let $m_2 = 1$ and $i_1 = i$, then inequality (22) becomes

$$x(\delta(S)) \geq 2\left(\sum_{u\in V\setminus S}\sum_{k\in K} w_{iu}^k + \sum_{l=1}^{m_1}\sum_{(u,k)\in S_l} w_{j_l u}^k\right).$$

The right-hand side of this inequality is equal to $2(z(i : V \setminus S) + \sum_{l=1}^{m_1}\sum_{(u,k)\in S_l} w_{j_l u}^k)$ and is greater than or equal to $2z(i : V \setminus S)$. Hence these inequalities dominate constraints (12). ∎

Finally, the third family of inequalities exploits the capacity limitation as it is done in the vehicle routing problem.

Theorem 3. *Consider a nonempty set $S \subset V$, a partition $(S_1, \ldots, S_m)$ of $S \times K$ and distinct nodes $j_1, \ldots, j_m$ be in $V \setminus S$. Inequalities*

$$x(\delta(S)) \geq 2\left(\left\lceil \frac{|S|}{q-1}\right\rceil - \sum_{i\in S} z_{ii} - \sum_{i\in S}\sum_{k\in K} w_{ii}^k + \sum_{l=1}^{m}\sum_{(u,k)\in S_l} w_{j_l u}^k\right) \quad (23)$$

and

$$x(\delta(S)) \geq 2\left(\left\lceil \frac{|S|}{q-1}\right\rceil - \frac{\sum_{i\in S} z_{ii}}{q-1} - 1 - \sum_{i\in S}\sum_{k\in K} w_{ii}^k + \sum_{l=1}^{m}\sum_{(u,k)\in S_l} w_{j_l u}^k\right) \quad (24)$$

are valid.

Proof. When there are $\sum_{i\in S} z_{ii}$ hubs in set S, we need at least $\left\lceil \frac{|S|-\sum_{i\in S} z_{ii}}{q-1}\right\rceil$ rings to cover the nodes in set S. As there are $\sum_{i\in S}\sum_{k\in K} w_{ii}^k$ rings adjacent to hubs in set S, the remaining $\left\lceil \frac{|S|-\sum_{i\in S} z_{ii}}{q-1}\right\rceil - \sum_{i\in S}\sum_{k\in K} w_{ii}^k$ rings are adjacent to hubs in set $V \setminus S$. There are also at least $\sum_{l=1}^{m}\sum_{(u,k)\in S_l} w_{j_l u}^k$ distinct rings adjacent to hubs in set S serving nodes in $V \setminus S$. Hence overall at least $2\left(\left\lceil \frac{|S|-\sum_{i\in S} z_{ii}}{q-1}\right\rceil - \sum_{i\in S}\sum_{k\in K} w_{ii}^k + \sum_{l=1}^{m}\sum_{(u,k)\in S_l} w_{j_l u}^k\right)$ access edges cross the cut $\delta(S)$. Finally, $\left\lceil \frac{|S|-\sum_{i\in S} z_{ii}}{q-1}\right\rceil \geq \left\lceil \frac{|S|}{q-1}\right\rceil - \left\lceil \frac{\sum_{i\in S} z_{ii}}{q-1}\right\rceil \geq \left\lceil \frac{|S|}{q-1}\right\rceil - \sum_{i\in S} z_{ii}$. Also, $\left\lceil \frac{|S|}{q-1}\right\rceil - \left\lceil \frac{\sum_{i\in S} z_{ii}}{q-1}\right\rceil \geq \left\lceil \frac{|S|}{q-1}\right\rceil - \frac{\sum_{i\in S} z_{ii}}{q-1} - 1$. ∎

4. BRANCH-AND-CUT ALGORITHM

As the formulation we presented for the ring/κ-rings problem involves an exponential number of constraints, we propose a branch-and-cut algorithm to solve it. In addition, we strengthen the Mixed Integer Programming (MIP) model with the valid inequalities presented in the previous section. The branch-and-cut approach for integer programming problems combines a branch-and-bound method for exploring a decision tree and a cutting plane method for computing bounds. At each node of the search tree, the cutting plane method improves a linear relaxation of the problem. When this is not further possible, the branch-and-bound algorithm proceeds. We outline next the main ingredients of the algorithm.

4.1. Initialization

To start the optimization, we initialize the linear program (LP) model by removing the exponential number of constraints (11) and (12) and the dominated constraints (13), and relaxing the integrality constraints on the variables of the original formulation. We keep constraints (14) and (15) for computational reasons, as they speed up the convergence of the method. Hence, the initial LP model is (1)–(10), (14)–(15), and the continuous relaxation of (16)–(18).

4.2. Cutting Plane Phase

Given an optimal LP solution (x^*, y^*, z^*, w^*) of the relaxed ring/ κ-rings problem, the separation routines for constraints (19), (11), and (12) are applied, in this sequence. The cutting plane phase is performed each time an integer solution is found, to check whether it is feasible, and each 10 branch-and-cut nodes. Moreover, the number of violated cuts of each family added to the model is limited to 20, and the total number of cuts added in each cut generation step is limited to 75. These limits are set to improve the general performance of the algorithm, both in terms of consumed memory (size of the subproblems) and time (computational cost of the separation procedures).

Inequalities (19) can be separated in polynomial time, as stated in the following result.

Proposition 1. *Inequalities (19) can be separated in $O(n^3\kappa)$ time.*

Proof. It suffices to observe that for a given edge $\{i,j\} \in E$, the right-hand side of inequality (19) is minimized by letting $K_l = \left\{k \in K : w_{il}^{*k} \leq w_{jl}^{*k}\right\}$ for all $l \in V \setminus \{i,j\}$. ∎

The subtour elimination constraints for the Traveling Salesman Problem (TSP) can be separated in polynomial time by solving a max-flow/min-cut problem on an appropriately defined support graph. We follow the same idea to devise exact separation procedures for inequalities (11) and (12). The detailed algorithms are given in [25].

Finally, constraints (22), (23), and (24) are not directly separated, but their violation is checked each time a violated inequality (12) is found. More precisely, for a given $S \subset V$, the violation test for (22) is done by checking whether

$$x^*(\delta(S)) < 2\sum_{k\in K}\left(\sum_{u\in V\setminus S}\max_{i\in S} w_{iu}^{*k} + \sum_{u\in S}\max_{j\in V\setminus S} w_{ju}^{*k}\right).$$

TABLE 1. Results for the ring/1-ring network design problem

			Class I					Class II				
f_j	n	q	*opt*	*%-gap*	*cpu*	*nodes*	*nCuts*	*opt*	*%-gap*	*cpu*	*nodes*	*nCuts*
[1, 500]	15	$\lceil 3n/4 \rceil$	910	0.00	0.37	0	122	1,462	0.00	0.25	0	158
		$\lceil n/2 \rceil$	913	0.25	0.36	7	127	1,466	0.59	0.20	8	144
		$\lceil n/4 \rceil$	1,105	4.90	1.33	60	349	1,792	0.85	2.15	7	393
	20	$\lceil 3n/4 \rceil$	1,216	0.29	0.78	9	213	1,323	1.14	0.97	7	315
		$\lceil n/2 \rceil$	1,216	1.15	0.14	12	98	1,323	0.98	2.68	92	516
		$\lceil n/4 \rceil$	1,359	4.13	6.72	205	715	1,713	6.04	102.95	753	2,314
	30	$\lceil 3n/4 \rceil$	979	0.00	0.75	0	260	1,321	0.45	1.11	5	362
		$\lceil n/2 \rceil$	979	0.06	1.06	18	284	1,336	0.11	2.20	4	330
		$\lceil n/4 \rceil$	1,139	3.60	58.06	863	1,431	1,653	6.89	490.37	1,208	6,137
	40	$\lceil 3n/4 \rceil$	737	0.77	2.42	33	409	1,435	0.79	15.29	145	1,184
		$\lceil n/2 \rceil$	737	0.77	3.26	29	285	1,438	0.20	2.98	7	478
		$\lceil n/4 \rceil$	852	2.67	2,668.97	7,122	6,658	1,715	5.28	2,994.88	2,318	11,122
[500, 1000]	15	$\lceil 3n/4 \rceil$	2,591	0.00	0.11	0	59	2,752	0.00	0.03	0	40
		$\lceil n/2 \rceil$	2,591	0.00	0.09	0	76	2,752	0.00	0.02	0	48
		$\lceil n/4 \rceil$	3,284	0.38	0.30	3	263	3,581	3.93	2.59	52	660
	20	$\lceil 3n/4 \rceil$	2,706	0.00	0.20	0	122	2,762	0.00	0.14	0	73
		$\lceil n/2 \rceil$	2,706	0.16	0.44	6	178	2,806	1.22	5.09	201	750
		$\lceil n/4 \rceil$	3,497	0.97	10.81	139	881	3,775	2.67	49.84	485	2,264
	30	$\lceil 3n/4 \rceil$	2,560	0.00	0.47	0	129	2,735	0.00	0.83	0	349
		$\lceil n/2 \rceil$	2,560	0.00	0.25	0	129	2,763	0.37	7.60	101	840
		$\lceil n/4 \rceil$	3,262	5.33	70.64	725	2,261	3,564	5.54	446.88	924	5,917
	40	$\lceil 3n/4 \rceil$	2,464	0.01	1.59	4	306	2,518	0.89	6.60	130	792
		$\lceil n/2 \rceil$	2,464	0.02	1.68	4	297	2,536	1.03	103.13	849	2,582
		$\lceil n/4 \rceil$	3,207	1.79	513.66	2,118	3,853	3,400	3.54	4,227.63	4,676	11,049
[1200, 1700]	15	$\lceil 3n/4 \rceil$	4,510	0.00	0.27	0	140	5,062	0.04	0.23	3	137
		$\lceil n/2 \rceil$	4,513	0.07	0.28	5	133	5,066	0.08	0.20	5	140
		$\lceil n/4 \rceil$	5,905	5.13	8.17	178	1,241	6,592	0.20	0.90	9	386
	20	$\lceil 3n/4 \rceil$	4,816	0.00	1.90	0	227	4,923	0.34	1.11	12	405
		$\lceil n/2 \rceil$	4,816	0.00	0.27	0	96	4,923	0.26	2.57	72	700
		$\lceil n/4 \rceil$	6,159	0.91	15.90	383	1,102	6,513	1.57	133.10	1,157	2,342
	30	$\lceil 3n/4 \rceil$	4,579	0.00	0.51	0	226	4,677	0.35	1.59	26	408
		$\lceil n/2 \rceil$	4,579	0.22	1.36	41	328	4,682	0.92	4.01	105	831
		$\lceil n/4 \rceil$	5,939	0.44	79.26	1,106	1,707	6,215	2.39	1,337.43	2,685	7,260
	40	$\lceil 3n/4 \rceil$	4,337	0.14	1.92	37	427	5,035	0.23	13.43	110	1,050
		$\lceil n/2 \rceil$	4,337	0.14	4.45	64	361	5,038	0.08	2.31	6	486
		$\lceil n/4 \rceil$	5,652	0.41	2,692.14	8,169	6,377	6,515	1.39	2,573.95	2,842	11,007

If so, we have found a violated constraint (22) for the given implicit partition.

Similarly, for a given $S \subset V$, the violation tests for (23) and (24) consist in checking, respectively, whether

$$x^*(\delta(S)) < 2\left(\left\lceil \frac{|S|}{q-1} \right\rceil - \sum_{i\in S} z^*_{ii} - \sum_{i\in S}\sum_{k\in K} w^{*k}_{ii} + \sum_{u\in S}\sum_{k\in K} \max_{j\in V\setminus S} w^{*k}_{ju}\right)$$

and

$$x^*(\delta(S)) < 2\left(\left\lceil \frac{|S|}{q-1} \right\rceil - \frac{\sum_{i\in S} z^*_{ii}}{q-1} - 1 - \sum_{i\in S}\sum_{k\in K} w^{*k}_{ii} + \sum_{u\in S}\sum_{k\in K} \max_{j\in V\setminus S} w^{*k}_{ju}\right).$$

4.3. *Branching Strategy*

To select the candidate variables for branching, we prioritize the set of variables z_{ii}. The reason is that setting $z_{ii} = 0$ or $z_{ii} = 1$ fixes several other variables to 0 and 1 as well. Among the candidate variables, we branch on the one whose value in the linear relaxation solution is furthest from being an integer, that is, is closest to 0.5. We branch on the other variables only when all variables z_{ii}'s are integer, and in that case we use the strong branching rule incorporated in CPLEX to select the next branch-and-bound tree node to explore.

5. COMPUTATIONAL RESULTS

The branch-and-cut algorithm was implemented in C++ and run on a personal computer with an Intel Core i7 CPU at 3.4 GHz and 16 GB of RAM. We used CPLEX 12.5 as mixed integer linear programming solver, keeping the default settings except for the branching rule.

TABLE 2. Results for the ring/2-rings network design problem

			Class I					Class II				
f_j	n	q	*opt*	*%-gap*	*cpu*	*nodes*	*nCuts*	*opt*	*%-gap*	*cpu*	*nodes*	*nCuts*
[1, 500]	15	$\lceil 3n/4 \rceil$	910	0.00	0.25	0	97	1,462	0.00	0.17	0	45
		$\lceil n/2 \rceil$	913	0.00	0.19	0	77	1,466	0.00	1.64	0	137
		$\lceil n/4 \rceil$	963	3.76	1.84	79	239	1,538	2.59	2.07	35	230
	20	$\lceil 3n/4 \rceil$	1,216	1.18	0.34	24	129	1,323	1.35	0.95	28	202
		$\lceil n/2 \rceil$	1,216	0.00	0.70	0	162	1,323	1.03	1.47	10	214
		$\lceil n/4 \rceil$	1,301	4.53	42.59	594	1,272	1,502	5.72	104.12	1,533	1,807
	30	$\lceil 3n/4 \rceil$	979	0.00	0.36	0	44	1,321	0.58	1.23	6	221
		$\lceil n/2 \rceil$	979	0.00	0.55	0	100	1,336	1.62	7.86	91	588
		$\lceil n/4 \rceil$	1,026	3.45	680.43	4,888	2,881	1,480	6.43 (0.46)	t.l.	16,583	7,701
	40	$\lceil 3n/4 \rceil$	737	0.81	2.07	32	326	1,435	1.39	175.49	2,213	1,817
		$\lceil n/2 \rceil$	737	0.81	6.16	79	436	1,438	0.66	83.57	400	1,975
		$\lceil n/4 \rceil$	758	2.94	470.20	1,593	2,817	1,655	10.55 (8.16)	t.l.	2,514	14,870
[500, 1000]	15	$\lceil 3n/4 \rceil$	2,591	0.00	0.08	0	25	2,752	0.00	0.06	0	27
		$\lceil n/2 \rceil$	2,591	0.00	0.06	0	25	2,752	0.00	0.05	0	18
		$\lceil n/4 \rceil$	2,721	2.61	4.59	277	412	2,799	0.32	0.33	12	106
	20	$\lceil 3n/4 \rceil$	2,706	0.46	0.27	18	84	2,762	0.00	0.14	0	30
		$\lceil n/2 \rceil$	2,706	0.14	0.72	12	158	2,806	1.56	4.91	205	431
		$\lceil n/4 \rceil$	2,799	2.22	11.89	272	606	2,959	3.04	114.41	2,204	1,674
	30	$\lceil 3n/4 \rceil$	2,560	0.00	0.73	0	97	2,735	0.00	0.87	0	201
		$\lceil n/2 \rceil$	2,560	0.00	0.59	0	86	2,763	0.54	21.42	226	1,026
		$\lceil n/4 \rceil$	2,581	0.35	31.22	244	976	2,929	3.97 (1.37)	t.l.	13,333	7,646
	40	$\lceil 3n/4 \rceil$	2,464	0.04	5.82	50	330	2,518	1.06	132.82	1,333	1,694
		$\lceil n/2 \rceil$	2,464	0.03	5.01	20	317	2,536	1.04	175.14	1,581	1,546
		$\lceil n/4 \rceil$	2,519	1.50	386.91	1,558	2,733	2,744	6.15 (4.38)	t.l.	5,001	11,717
[1200, 1700]	15	$\lceil 3n/4 \rceil$	4,510	0.00	0.19	0	85	5,062	0.00	1.42	0	38
		$\lceil n/2 \rceil$	4,513	0.00	1.50	0	80	5,066	0.08	0.30	9	79
		$\lceil n/4 \rceil$	4,563	0.76	1.83	81	291	5,138	0.65	1.76	44	289
	20	$\lceil 3n/4 \rceil$	4,816	0.30	0.25	23	100	4,923	0.35	1.51	17	219
		$\lceil n/2 \rceil$	4,816	0.28	0.30	16	152	4,923	0.39	1.89	41	300
		$\lceil n/4 \rceil$	4,901	0.99	9.72	161	847	5,102	1.68	201.83	2,300	2,703
	30	$\lceil 3n/4 \rceil$	4,579	0.00	0.34	0	40	4,677	0.38	7.69	188	531
		$\lceil n/2 \rceil$	4,579	0.00	0.41	0	65	4,682	0.02	1.87	4	307
		$\lceil n/4 \rceil$	4,626	0.77	896.48	6,724	3,469	4,849	1.54	868.75	3,552	4,478
	40	$\lceil 3n/4 \rceil$	4,337	0.14	2.11	39	297	5,035	0.39	239.49	2,307	2,032
		$\lceil n/2 \rceil$	4,337	0.16	18.16	141	686	5,038	0.19	88.17	487	2,077
		$\lceil n/4 \rceil$	4,358	0.52	1,035.02	2,901	3,912	5,171	2.02 (0.61)	t.l.	8,394	12,056

We tested the algorithm on two data sets used in [25] consisting of 36 instances each, with a number of nodes ranging from 15 to 40. In the *Class I* instances, edge costs c_{ij} are randomly generated in $[1, 100]$. In the instances of the *Class II* node coordinates are randomly generated in $[0, 100] \times [0, 100]$, and the edge costs c_{ij} are computed as the Euclidean distance between the points i and j. In both types of instances, we define the access and backbone link costs as $a_e = c_e$ and $b_e = 4c_e$, respectively. That is, the cost of installing a backbone link is four times higher than the cost of installing an access link on the same edge. The hub capacity q takes values in $\{\lceil 3n/4 \rceil, \lceil n/2 \rceil, \lceil n/4 \rceil\}$, thus going from a lax to a tight value. For each combination of number of nodes and capacity, we tried three different settings for the hub fixed costs, randomly generating f_j in $[1, 500]$, $[500, 1000]$ or $[1200, 1700]$.

Tables 1–2, and 3 show the results of our experiments with the branch-and-cut algorithm for the three particular cases of the ring/ κ-ring problem where $\kappa = 1$, $\kappa = 2$, and $\kappa = 3$, respectively. For each instance, we report in the different columns the range for the hub fixed costs (f_j), the number of nodes (n), the hub capacity (q), the optimal value (*opt*), the duality gap, that is, the percentage gap between the optimal value and the lower bound at the end of the root node (*%-gap*), the total computation time in seconds (*cpu*), the number of nodes explored in the branch-and-cut tree (*nodes*), and the total number of cuts generated (*nCuts*). We imposed a time limit of two hours. If the time limit is reached before proving optimality, we write "t.l." in column *cpu*, use the objective function value of the best solution found instead of the optimal value to compute the duality gap, and report in brackets the percentage gap between the value of the best solution found and the lower bound at the end of the 2 hours.

All the instances in Table 1 are solved to optimality within the time limit. The instances of Class II, with Euclidean distances, seem to be more difficult. In fact, 11 out of the 36 instances of Class I are solved without branching (i.e., at the root node of the branch-and-cut tree), while that figure goes down to 5 for Class II. The hardest instances are those with 40 nodes and the tightest capacity ($q = \lceil n/4 \rceil$). When looking

TABLE 3. Results for the ring/3-rings network design problem

			Class I instances					Class II instances				
f_j	n	q	*opt*	*%-gap*	*cpu*	*nodes*	*nCuts*	*opt*	*%-gap*	*cpu*	*nodes*	*nCuts*
[1, 500]	15	$\lceil 3n/4 \rceil$	910	0.00	0.28	0	97	1,462	0.00	0.33	0	80
		$\lceil n/2 \rceil$	913	0.30	0.33	5	89	1,466	0.00	0.30	0	80
		$\lceil n/4 \rceil$	963	4.09	5.09	113	612	1,538	1.97	7.83	97	649
	20	$\lceil 3n/4 \rceil$	1,216	0.33	0.36	4	103	1,323	1.25	2.95	17	216
		$\lceil n/2 \rceil$	1,216	0.19	0.75	4	169	1,323	1.68	4.96	105	515
		$\lceil n/4 \rceil$	1,301	4.79	260.23	1,755	3,607	1,501	7.85	382.97	3,357	3,574
	30	$\lceil 3n/4 \rceil$	979	0.00	0.34	0	52	1,321	1.74	4.09	47	395
		$\lceil n/2 \rceil$	979	0.10	2.45	24	417	1,336	0.85	45.15	174	1,764
		$\lceil n/4 \rceil$	1,026	3.16	1,852.54	5,730	5,875	1,463	6.25 (0.77)	t.l.	7,127	11,939
	40	$\lceil 3n/4 \rceil$	737	0.86	3.23	59	530	1,435	1.39	468.02	2,279	2,957
		$\lceil n/2 \rceil$	737	0.85	4.85	53	364	1,438	0.83	83.60	532	992
		$\lceil n/4 \rceil$	772	4.72 (3.51)	t.l.	3,531	10,421	1,798	18.88 (17.47)	t.l.	1,202	16,913
[500, 1000]	15	$\lceil 3n/4 \rceil$	2,591	0.00	0.11	0	25	2,752	0.00	0.11	0	27
		$\lceil n/2 \rceil$	2,591	0.00	0.12	0	27	2,752	0.00	0.05	0	36
		$\lceil n/4 \rceil$	2,721	2.69	10.34	325	610	2,799	0.29	2.89	18	50
	20	$\lceil 3n/4 \rceil$	2,706	0.14	0.33	11	88	2,762	0.00	0.09	0	35
		$\lceil n/2 \rceil$	2,706	0.00	0.42	0	142	2,806	1.55	10.76	294	713
		$\lceil n/4 \rceil$	2,799	2.56	81.96	1,375	1,502	2,959	3.36	1,111.90	6,693	4,617
	30	$\lceil 3n/4 \rceil$	2,560	0.00	0.81	0	101	2,735	0.00	0.72	0	150
		$\lceil n/2 \rceil$	2,560	0.00	0.59	0	97	2,763	0.54	20.98	269	1,020
		$\lceil n/4 \rceil$	2,581	0.36	151.49	271	3,543	2,939	4.17 (2.27)	t.l.	7,858	10,062
	40	$\lceil 3n/4 \rceil$	2,464	0.04	2.98	6	173	2,518	1.06	60.64	745	1,512
		$\lceil n/2 \rceil$	2,464	0.08	4.57	28	310	2,536	1.06	221.83	1,384	2,722
		$\lceil n/4 \rceil$	2,519	1.54 (0.88)	t.l.	5,995	9,261	2,815	8.59 (7.46)	t.l.	1,897	14,232
[1200, 1700]	15	$\lceil 3n/4 \rceil$	4,510	0.00	0.22	0	86	5,062	0.00	0.19	0	76
		$\lceil n/2 \rceil$	4,513	0.12	0.33	14	82	5,066	0.09	0.28	7	92
		$\lceil n/4 \rceil$	4,563	0.78	2.89	101	499	5,138	0.94	2.39	51	323
	20	$\lceil 3n/4 \rceil$	4,816	0.06	0.53	4	160	4,923	0.39	2.07	18	317
		$\lceil n/2 \rceil$	4,816	0.09	1.31	7	258	4,923	0.45	5.83	113	713
		$\lceil n/4 \rceil$	4,901	1.15	38.42	327	1,637	5,101	1.98	2,491.74	12,669	6,599
	30	$\lceil 3n/4 \rceil$	4,579	0.00	0.94	0	143	4,677	0.38	8.89	198	643
		$\lceil n/2 \rceil$	4,579	0.00	0.64	0	121	4,682	0.00	1.92	0	360
		$\lceil n/4 \rceil$	4,626	0.80	3,199.47	9,375	6,484	4,852	2.00 (0.61)	t.l.	11,722	10,474
	40	$\lceil 3n/4 \rceil$	4,337	0.14	6.52	85	627	5,035	0.40	745.83	5,178	2,605
		$\lceil n/2 \rceil$	4,337	0.15	7.33	75	743	5,038	0.15	38.84	167	1,518
		$\lceil n/4 \rceil$	4,358	0.52 (0.12)	t.l.	7,425	7,274	5,151	1.39 (0.42)	t.l.	7,745	11,406

at a given node size, it is also clear that the instances with the most restrictive capacity are the hardest ones. In fact, the duality gaps are usually below 1% for the instances with larger capacities, and the computation times are also much smaller. Regarding the fixed costs for hubs, the duality gaps tend to decrease as they increase, this being more noticeable in Class II.

Table 2 shows the results for the ring/2-rings problem. In this case, the branch-and-cut algorithm fails to solve five instances of Class II within the time limit of 2 hours. They are all cases with 30 and 40 nodes and with the tightest capacity. On the contrary, the algorithm finds the optimal solution of all the instances with random distances (Class I). A total of 13 instances out of 36 in Class I and 7 in Class II are solved at the root node, and the duality gaps are again usually below or around 1% except when the capacity is the tightest.

The results for the ring/3-rings problem are given in Table 3. The algorithm does not manage to prove optimality on the three instances of Class I with 40 nodes and $q = \lceil n/4 \rceil$, and on the six instances of Class II with that capacity and 30 and 40 nodes. However, optimal solutions are obtained for all the other instances in few minutes. It is interesting to observe that, on our test bed, the optimal solutions of the ring/3-rings problem coincide in most of cases with the optimal solutions of the ring/2-rings problem. The exceptions are two instances of Class II with $n = 20$ and the tightest capacity.

TABLE 4. Average number of cuts added when solving the ring/2-rings problem

	f_j	(11)	(12)	(19)	(22)	(23)
Class I	[1, 500]	11.17	37.25	621.67	37.25	7.67
	[500, 1000]	1.42	26.17	428.83	26.17	4.83
	[1200, 1700]	10.92	56.75	699.75	56.75	11.17
Class II	[1, 500]	190.17	186.17	1,872.58	186.17	48.83
	[500, 1000]	70.50	261.08	1,511.67	261.08	72.00
	[1200, 1700]	39.33	219.50	1,559.67	219.50	54.42

Table 4 shows the average number of cuts of each type generated during the computation for the problems with $\kappa = 2$.

TABLE 5. Performance of the basics and complete branch-and-cut algorithms on the ring/2-rings problem

				basic1 B&C		*basic2 B&C*		*basic3 B&C*		*Complete B&C*	
	f_j	n	q	*%-gap*	*cpu*	*%-gap*	*cpu*	*%-gap*	*cpu*	*%-gap*	*cpu*
Class I	[1, 500]	15	$\lceil 3n/4 \rceil$	0.98	4.48	0.00	0.20	0.00	0.25	0.00	0.25
			$\lceil n/2 \rceil$	1.44	6.08	0.00	0.34	0.00	0.20	0.00	0.19
			$\lceil n/4 \rceil$	5.01	150.12	3.76	2.78	3.76	4.99	3.76	1.84
		20	$\lceil 3n/4 \rceil$	1.39	22.74	1.18	0.37	1.18	0.28	1.18	0.34
			$\lceil n/2 \rceil$	0.32	15.87	0.11	0.23	0.00	0.67	0.00	0.70
			$\lceil n/4 \rceil$	5.01	1,042.98	4.53	72.54	4.53	55.55	4.53	42.59
	[500, 1000]	15	$\lceil 3n/4 \rceil$	0.00	2.87	0.00	0.09	0.00	0.11	0.00	0.08
			$\lceil n/2 \rceil$	0.00	3.67	0.00	0.06	0.00	0.08	0.00	0.06
			$\lceil n/4 \rceil$	3.12	249.24	2.78	5.34	2.61	4.65	2.61	4.59
		20	$\lceil 3n/4 \rceil$	0.50	14.54	0.46	0.28	0.46	0.25	0.46	0.27
			$\lceil n/2 \rceil$	0.30	12.98	0.21	0.44	0.14	0.72	0.14	0.72
			$\lceil n/4 \rceil$	2.82	341.35	2.22	4.37	2.22	11.62	2.22	11.89
	[1200, 1700]	15	$\lceil 3n/4 \rceil$	0.19	5.57	0.00	0.20	0.00	0.22	0.00	0.19
			$\lceil n/2 \rceil$	0.10	8.83	0.06	0.27	0.00	2.22	0.00	1.50
			$\lceil n/4 \rceil$	1.11	44.01	0.76	1.54	0.76	1.90	0.76	1.83
		20	$\lceil 3n/4 \rceil$	0.32	25.19	0.30	0.28	0.30	0.37	0.30	0.25
			$\lceil n/2 \rceil$	0.29	18.6	0.28	0.42	0.28	0.31	0.28	0.30
			$\lceil n/4 \rceil$	1.01	1,248.91	1.01	6.04	0.99	9.42	0.99	9.72
Class II	[1, 500]	15	$\lceil 3n/4 \rceil$	0.00	2.89	0.00	0.11	0.00	1.44	0.00	0.17
			$\lceil n/2 \rceil$	0.85	4.59	0.24	0.36	0.24	0.36	0.00	1.64
			$\lceil n/4 \rceil$	2.59	25.83	2.59	1.06	2.59	2.57	2.59	2.07
		20	$\lceil 3n/4 \rceil$	1.65	15.85	1.47	1.97	1.40	1.44	1.35	0.95
			$\lceil n/2 \rceil$	1.80	29.92	1.57	1.47	1.22	2.31	1.03	1.47
			$\lceil n/4 \rceil$	16.75 (13.18)	t.l.	10.48 (0.92)	t.l.	10.38	2,505.27	5.72	104.12
	[500, 1000]	15	$\lceil 3n/4 \rceil$	0.00	1.42	0.00	0.05	0.00	0.05	0.00	0.06
			$\lceil n/2 \rceil$	0.00	1.08	0.00	0.05	0.00	0.06	0.00	0.05
			$\lceil n/4 \rceil$	0.32	6.46	0.32	1.48	0.32	0.31	0.32	0.33
		20	$\lceil 3n/4 \rceil$	0.00	21.79	0.00	0.08	0.00	0.19	0.00	0.14
			$\lceil n/2 \rceil$	1.56	97.97	1.56	5.48	1.56	4.96	1.56	4.91
			$\lceil n/4 \rceil$	5.92 (3.10)	t.l.	5.05	991.20	5.00	520.06	3.04	114.41
	[1200, 1700]	15	$\lceil 3n/4 \rceil$	0.00	6.66	0.00	0.16	0.00	0.16	0.00	1.42
			$\lceil n/2 \rceil$	0.27	4.27	0.16	0.25	0.15	0.25	0.08	0.30
			$\lceil n/4 \rceil$	0.81	53.74	0.65	1.00	0.65	1.95	0.65	1.76
		20	$\lceil 3n/4 \rceil$	0.48	19.84	0.38	0.81	0.36	0.78	0.35	1.51
			$\lceil n/2 \rceil$	0.50	73.35	0.46	2.57	0.46	1.39	0.39	1.89
			$\lceil n/4 \rceil$	3.54 (1.61)	t.l.	3.04	1,143.64	3.03	548.80	1.68	201.83

We tested several different orders for adding violated constraints (12), (22), (23), and (24) to the model, as they are all found in the separation routine for (12). We eventually decided not to use inequalities (24) in the final version of the branch-and-cut algorithm, as they did not seem to add anything computationally with respect to constraints (23), independently of their order, on our instances, and chose to add (23), (12), and (22) in that sequence. For each hub fixed cost setting, we report in a single line the data corresponding to the 12 instances with that setting. The largest number of violated cuts generated corresponds to the valid inequalities (19). This may be explained by the fact that they are separated at the beginning of the cutting plane phase and that they replace constraints (13), which are necessary to define the problem. In the second place come inequalities (12) and (22). Their numbers coincide because we check the violation of constraints (22) inside the separation routine for (12), as said before, and we are certain to find a violated constraint of the first type when the latter is violated. However, this does not happen with constraints (23). Violated cuts of type (11) appear too, mainly in the first nodes of the branch-and-cut. Note also that the number of violated cuts found when solving the instances of Class II is larger than the number of violated cuts generated when solving instances of Class I. This again indicates the greater difficulty of the instances with Euclidean distances.

Finally, to evaluate the effect of the valid inequalities used in our branch-and-cut approach, we performed an experiment consisting of comparing the complete algorithm described in Section 4 with three other versions of it. The first version (*basic1 B&C*) is a basic algorithm that uses only the constraints (1)–(18) in the model. All constraints are incorporated to the initial LP, except the connectedness constrains (11) and (12) which are separated. In the second version (*basic2 B&C*), constraints (13) are removed from the initial LP, and instead the stronger constraints (19) are separated. The third version of the algorithm (*basic3 B&C*) incorporates the violation checking test for the valid inequalities (22). Table 5 displays the results obtained for the smallest instances, those with 15 and 20 nodes, when solving

the ring/2-rings problem. We report, for each algorithm, the gap of the linear programming relaxation at the end of the root node and the total computing time. It is clear from the results that the separation of the valid inequalities presented in Section 3 not only reduces the gap, but significantly reduces the computation times. In fact, the first basic branch-and-cut algorithm is unable to solve three of the instances with 20 nodes within the time limit of 2 hours, and takes substantially more time to solve each of the other instances than the complete branch-and-cut algorithm.

6. CONCLUSIONS

In this article, we addressed a two-level network design problem with ring topology in both levels of the network. This structure ensures network survivability, understood as protection against single-edge failures. We imposed that there is at least one and at most κ access ring networks adjacent to each hub. The number of nodes in each access ring is also limited. Solving this complex optimization problem implies to chose the location of the hubs, to decide on the assignment of the other nodes to the hubs, and to design the backbone and access rings.

We present a mathematical model and valid inequalities for the problem. Some of the inequalities generalize others given in the literature for related problems. We describe an exact branch-and-cut algorithm, and we show computational results on two classes of instances with up to 40 nodes and different features. The proposed algorithm is able to solve to optimality most of the instances in less than 15 min, which is a quite reasonable time.

The formulation and valid inequalities presented here may also be used for problems where rings are used for routing vehicles. We are interested, as future work, in extending these results to solve problems with multiple depots, customers with arbitrary demands, and vehicles with different capacities.

REFERENCES

[1] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani, Designing hierarchical survivable networks, Oper Res 46 (1998), 116–136.

[2] A. Balakrishnan, P. Mirchandani, and H.P. Natarajan, Connectivity upgrade models for survivable network design, Oper Res 57 (2009), 170–186.

[3] R. Baldacci, M. Dell'Amico, and J.J. Salazar-González, The capacitated m-ring star problem, Oper Res 55 (2007), 1142–1162.

[4] P. Carroll and S. McGarraghy, A decomposition algorithm for the ring spur assignment problem, Int Trans Oper Res 20 (2013), 119–139.

[5] I. Contreras and E. Fernández, General network design: A unified view of combined location and network design problems, Eur J Oper Res 219 (2012), 680–697.

[6] R. Cuda, G. Guastaroba, and M.G. Speranza, A survey on two-echelon routing problems, Comput Oper Res 55 (2015), 185–199.

[7] B. Fortz and M. Labbé, Two-connected networks with rings of bounded cardinality, Comput Optim Appl 27 (2014), 123–148.

[8] B. Fortz, M. Labbé, and F. Maffioli, Solving the two-connected network with bounded meshes problem, Oper Res 48 (2000), 866–877.

[9] B. Fortz, A.R. Mahjoub, S. McCormick, and P. Pesneau, Two-edge connected subgraphs with bounded rings: Polyhedral results and branch-and-cut, Math Program 105 (2006), 85–111.

[10] B. Fortz, P. Soriano, and C. Wynants, A Tabu search algorithm for self-healing ring network design, Eur J Oper Res 151 (2003), 280–295.

[11] P. Fouilhoux, O.E. Karasan, A.R. Mahjoub, O. Ozkok, and H. Yaman, Survivability in hierarchical telecommunications networks, Networks 59 (2012), 37–58.

[12] E. Gourdin, M. Labbé, and H. Yaman, "Telecommunication and location," Facility location: Applications and theory, Z. Drezner and H.W. Hamacher (Editors), Springer-Verlag, Berlin - Heidelberg, 2002, pp. 275–305.

[13] M. Grötschel, C.L. Monma, and M. Stoer, "Design of survivable networks," Handbooks in OR & MS, Vol. 7, Network models, M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (Editors), North-Holland, Amsterdam, The Netherlands, 1995, pp. 617–671.

[14] O. Karasan, A.R. Mahjoub, O. Ozkok, and H. Yaman, Survivability in hierarchical telecommunications networks under dual homing, INFORMS J Comput 26 (2014), 1–15.

[15] H. Kerivin and A.R. Mahjoub, Design of survivable networks: A survey, Networks 46 (2005), 1–21.

[16] H. Kerivin and A.R. Mahjoub, On survivable network polyhedra, Discrete Math 290 (2005), 183–210.

[17] J.G. Klincewicz, Hub location in backbone/tributary network design: A review, Location Sci 6 (1998), 307–335.

[18] M. Labbé, I. Rodríguez-Martín, and J.J. Salazar-González, A branch-and-cut algorithm for the plant-cycle location problem, J Oper Res Soc 55 (2004), 513–520.

[19] M. Labbé, G. Laporte, I. Rodríguez-Martín, and J.J. Salazar-González, The ring star problem: Polyhedral analysis and exact algorithm, Networks 43 (2004), 177–189.

[20] C.Y. Lee and S.J. Koh, A design of the minimum cost ring-chain network with dual-homing survivability: A Tabu search approach, Comput Oper Res 24 (1997), 883–897.

[21] T.L. Magnanti and S. Raghavan, Strong formulations for network design problems with connectivity requirements, Networks 45 (2005), 61–79.

[22] A.R. Mahjoub, Two-edge connected spanning subgraphs and polyhedra, Math Program 64 (1994), 199–208.

[23] A.R. Mahjoub and P. Pesneau, On the Steiner 2-edge connected subgraph polytope, RAIRO Oper Res 42 (2008), 259–283.

[24] A. Proestki and M.C. Sinclair, Design and dimensioning of dual-homing hierarchical multi-ring networks, IEEE Proc Commun 47 (2000), 96–104.

[25] I. Rodríguez-Martín, J.J. Salazar-González, and H. Yaman, A branch-and-cut algorithm for two-level survivable network design problems, Comput Oper Res 67 (2016), 102–112.

[26] J.J. Shi and J.P. Fonseka, Hierarchical self-healing rings, IEEE ACM Trans Netw 3 (1995), 690–697.

[27] J.J. Shi and J.P. Fonseka, Analysis and design of survivable communications networks, IEEE Proc Commun 144 (1997), 322–330.

[28] P. Soriano, C. Wynants, R. Seguin, M. Labbé, M. Gendreau, and B. Fortz, "Design and dimensioning of survivable SDH/SONET networks," Telecommunications network planning, B. Sanso and P. Sariano (Editors), Springer, New York, 1999, pp. 147–167.

[29] M. Stoer, Design of survivable networks, Lecture notes in mathematics 1531, Springer-Verlag, Berlin - Heidelberg, 1992.

[30] T. Thomadsen and T. Stidsen, Hierarchical ring network design using branch-and-price, Telecommun Syst 29 (2005), 61–76.

[31] D. Vandenbussche and G.L. Nemhauser, The 2-edge-connected subgraph polyhedron, J Combin Optim 9 (2005), 357–379.