The Ring Star Problem: Polyhedral Analysis and Exact Algorithm

Martine Labbé
ISRO and SMG, Université Libre de Bruxelles, C.P. 210/01, B-1050, Brussels, Belgium

Gilbert Laporte
Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Inmaculada Rodríguez Martín and Juan José Salazar González
DEIOC, Facultad de Matemáticas, Universidad de La Laguna, 38271 La Laguna, Tenerife, Spain

In the Ring Star Problem, the aim is to locate a simple cycle through a subset of vertices of a graph with the objective of minimizing the sum of two costs: a ring cost proportional to the length of the cycle and an assignment cost from the vertices not in the cycle to their closest vertex on the cycle. The problem has several applications in telecommunications network design and in rapid transit systems planning. It is an extension of the classical location–allocation problem introduced in the early 1960s, and closely related versions have been recently studied by several authors. This article formulates the problem as a mixed-integer linear program and strengthens it with the introduction of several families of valid inequalities. These inequalities are shown to be facet-defining and are used to develop a branch-and-cut algorithm. Computational results show that instances involving up to 300 vertices can be solved optimally using the proposed methodology. © 2004 Wiley Periodicals, Inc.

Keywords: median cycle; branch and cut; telecommunications

1. INTRODUCTION

In a typical telecommunications network, the traffic is gathered from many sources, progressively combined in order to fill links of increasing capacity, and finally forwarded to its destination. Hence, most telecommunications networks are naturally structured in a multilayer hierarchical structure.

Although real telecommunications (voice, video, or data, nationwide or international networks) are usually structured into many such levels, most design problems considered by telecommunications companies concern only a part of the overall network. A generic telecommunications network consists of access networks which connect the terminals (users) to concentrators (switches or multiplexers) and a backbone network which interconnects these concentrators or connects them to a central unit (root). Telecommunications networks differ in the design of the access and backbone networks (see, e.g., Gourdin et al. [8] or Klincewicz [11]).

A solution to the design of some telecommunications networks is to connect terminals to concentrators by point-to-point links, which results in a star topology, and to interconnect the concentrators through a ring structure. This is the case in Digital Data Service (DDS) design, as mentioned, for example, in Xu et al. [18]. Roughly speaking, the problem then consists of selecting a subset of user locations where concentrators will be installed, interconnect them by a ring network, and assign the other user locations to those concentrators. The total cost of all connections must be minimized. Figure 1 illustrates a potential solution, where the solid lines represent the *internet* (ring) and the dashed lines represent the *intranet* (assignments). This problem was introduced in Moreno Pérez et al. [14] and solved heuristically.

Similar location–allocation problems arise in a number of planning contexts (see Cooper [4]). They consist of locating a number of facilities among the vertices of a graph

Received December 2001; accepted November 2003
Correspondence to: G. Laporte; e-mail: gilbert@crt.umontreal.ca
Contract grant sponsor: Canadian Natural Sciences and Engineering Research Council; contract grant number: OGP0039682
Contract grant sponsor: Coopération scientifique et technologique, Québec–Communauté française de Belgique
Contract grant sponsor: Ministerio de Ciencia y Technología; contract grant number: TIC2000-1750-CO6-02
Contract grant sponsor: Ministerio de Educación, Cultura y Deportes; contract grant number: SAB2000-0069

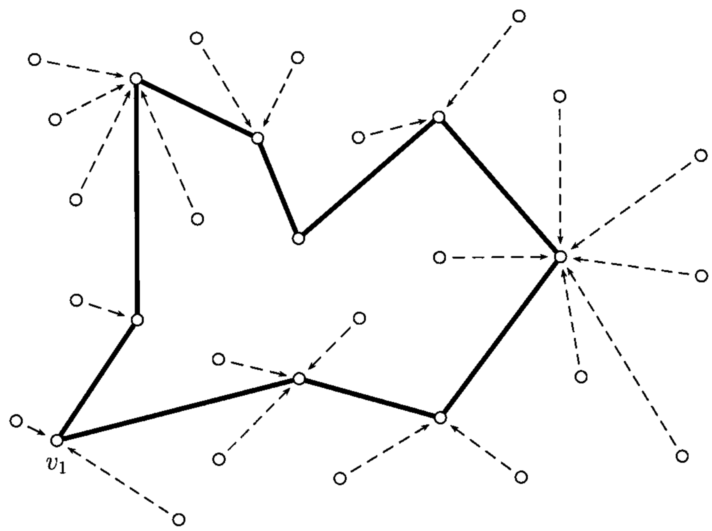

FIG. 1. Ring-star solution.

so that the allocation costs (i.e., the sum of distances from the remaining vertices to their closest facility) is minimized. In some applications, as in rapid transit systems planning, the facilities must be served on a single-vehicle route. Thus, in addition to the allocation cost, the objective includes the cost of a shortest Hamiltonian cycle visiting all the facilities. See Labbé et al. [12] for a review of applications and a more general class of problems consisting of locating structures in a graph.

In this article, we consider a problem common to both contexts, telecommunications and location–allocation, named the *Ring Star Problem* (RSP). The problem can be formally stated as follows: Consider a mixed graph $G = (V, E \cup A)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the vertex set, $E = \{[v_i, v_j] : v_i, v_j \in V, i < j\}$ is the edge set, and $A = \{(v_i, v_j) : v_i, v_j \in V\}$ is the arc set [loops (v_i, v_i) are included in A]. We assume that $n \geq 5$ to eliminate degenerate and trivial instances. Vertex v_1 is referred to as the *root* (or *depot*). Edges in E refer to the undirected connections in the internet, and arcs in A refer to the directed assignments in the intranet. There is a nonnegative *ring cost* c_{ij} associated with each edge $[v_i, v_j]$ and a nonnegative *assignment cost* d_{ij} associated with each arc (v_i, v_j). A solution to the RSP is a simple cycle through a subset V' of V including v_1 and at least two other vertices. The ring cost of a solution is the sum of the ring costs of all edges on the cycle. The assignment cost of a solution is defined as

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij}. \tag{1}$$

The RSP consists of determining a solution for which the sum of ring and assignment costs is minimized. The problem is $\mathcal{NP}$-hard since the special case in which the assignment costs are very large compared to the ring cost is the classical *Traveling Salesman Problem* (TSP).

Lee et al. [13] defined a very closely related problem (referred to as the *Steiner Ring Star Problem*) by considering an additional set of vertices W (representing *terminals* in telecommunications and *customers* in location–allocation) and setting the assignment cost to

$$\sum_{v_i \in W} \min_{v_j \in V'} d_{ij}. \tag{2}$$

They developed a branch and cut that solved instances with $|V| \leq 50$, $|W| \leq 90$, and $|V| + |W| \leq 100$. Xu et al. [18] proposed a tabu search algorithm for this problem and tested it on instances with $|V| \leq 300$ and $|W| \leq 300$. In addition, Current and Schilling [5] and Gendreau et al. [7] presented algorithms for variants of the RSP in which the ring cost is minimized subject to an upper bound on the cost of the largest assignment.

Clearly, practical applications of this type of models, particularly in telecommunications, contain additional constraints. For instance, a ring vertex should have a limit on the number of incoming external vertices or on the traffic coming from these vertices. Also, hybrid coax-fiber networks allow more sophisticated structures for the interconnection of the external vertices, as opposed to direct connections through ring vertices. The RSP is an interesting simplified version of these more realistic situations since it contains routing and location structures, two well-known hard combinatorial problems. Therefore, good cutting-plane algorithms for the RSP could prove valuable to approximate the more realistic problems. This is the case of a restricted RSP version in which the total ring cost is minimized while the total assignment cost cannot exceed a given threshold (see Rodríguez Martín [17]).

Our aim was to develop a polyhedral-based exact algorithm for the RSP. The remainder of this article is organized as follows: In Section 2, the RSP is formulated as a mixed-integer linear program to which several classes of valid inequalities are incorporated. In Section 3, we derive dimension and facet-defining results for the RSP. A branch-and-cut algorithm is described in Section 4. Section 5 presents extensive computational results on three classes of instances, one of them generated so as to compare our results with the ones in Lee et al. [13]. Conclusions follow in Section 6.

2. MATHEMATICAL MODEL

The RSP can be formulated as a mixed-integer linear program as follows: For each edge $[v_i, v_j] \in E$, let x_{ij} be a binary variable equal to 1 if and only if edge $[v_i, v_j]$ appears on the cycle. For each arc $(v_i, v_j) \in A$, let y_{ij} be a binary variable equal to 1 if and only if vertex v_i is assigned to vertex v_j on the cycle. If a vertex v_i is on the cycle, it is then assigned to itself, that is, $y_{ii} = 1$. In addition, for $S \subset V$, define $E(S) := \{[v_i, v_j] \in E : v_i, v_j \in S\}$ and $\delta(S) := \{[v_i, v_j] \in E : v_i \in S, v_j \notin S.\}$ If $S = \{v_i\}$, we simply write $\delta(i)$ instead of $\delta(\{v_i\})$. For $E' \subseteq E$, define $x(E') := \sum_{[v_i,v_j]\in E'} x_{ij}$.

The formulation is then

$$\text{minimize} \sum_{[v_i,v_j]\in E} c_{ij}x_{ij} + \sum_{(v_i,v_j)\in A} d_{ij}y_{ij}, \tag{3}$$

subject to

$$x(\delta(i)) = 2y_{ii} \qquad \text{for all } v_i \in V, \tag{4}$$

$$\sum_{v_j\in V} y_{ij} = 1 \qquad \text{for all } v_i \in V\backslash\{v_1\}, \tag{5}$$

$$x(\delta(S)) \geq 2 \sum_{v_j\in S} y_{ij} \qquad \text{for all } S \subset V : v_1 \notin S, v_i \in S, \tag{6}$$

$$x_{ij} \in \{0, 1\} \qquad \text{for all } [v_i, v_j] \in E, \tag{7}$$

$$y_{ij} \geq 0 \qquad \text{for all } (v_i, v_j) \in A, \tag{8}$$

$$y_{11} = 1 \tag{9}$$

$$y_{1j} = 0 \qquad \text{for all } v_j \in V\backslash\{v_1\}, \tag{10}$$

$$y_{jj} \text{ integer} \qquad \text{for all } v_j \in V\backslash\{v_1\}. \tag{11}$$

In this formulation, constraints (4) are degree constraints. They ensure that the degree of a vertex v_i is 2 if and only if it belongs to the cycle (i.e., $y_{ii} = 1$). Constraints (5) mean that either v_i is a vertex on the cycle (i.e., $y_{ii} = 1$) or v_i is assigned to a vertex v_j on the cycle (i.e., $y_{ii} = 0$ and $y_{ij} = 1$). Constraints (6) are connectivity constraints since they state that S must be connected to its complement by at least two edges of the cycle whenever at least one vertex $v_i \in S$ is assigned to $v_j \in S$. The combination of (4), (7), (9), and (10) guarantees that the solution will contain at least one cycle including the root. Constraints (6) limit the number of cycles to one by enforcing connectivity between vertices with $y_{ii} = 1$ and v_1; in other words, without constraints (6), the model would admit solutions like the one illustrated in Figure 2. The combination of (4), (5), (6), and (8) means that every vertex not belonging to the cycle is assigned to a vertex on the cycle. Integrality conditions on the y_{ij} variables ($v_i \neq v_j$) are unnecessary since, for a given integer x, the objective is minimized when vertices not on the cycle are assigned to the closest vertices on the cycle. Integer solutions are trivially determined in case of ties. Implicitly, the above model imposes that the cycle visits at least three vertices, including the root. Such a constraint guarantees that communication between vertices remains possible even if one vertex of the backbone network (e.g., the root) fails. Further, if reliability is unimportant, other potential solutions consisting of a degenerate cycle (i.e., only the root or only the root and another vertex) can easily be enumerated.

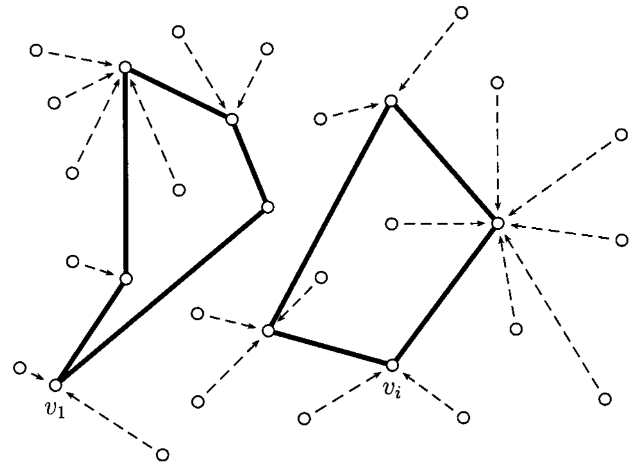

FIG. 2. Infeasible solution: constraint (6) is violated.

Observe that the model (3)–(11) allows the inclusion of a *hub-selection fixed-cost* d_{ii} for having vertex v_i in the cycle. Another important advantage of the model defined by (3)–(11) is that it can easily be extended to deal with hub capacity constraints. For instance, if vertex v_i in the cycle cannot serve more than l_i external vertices, then the following family of constraints is added to the model:

$$\sum_{(v_j,v_i)\in A} y_{ij} \leq l_i y_{ii} \qquad \text{for all } v_i \in V.$$

Clearly, the presence of these constraints can make an instance very difficult when compared to the uncapacitated version. Therefore, these additional constraints could require solution techniques different from those described in this article.

The linear relaxation of the model (3)–(11) can be strengthened through the introduction of additional valid inequalities. First, observe that connectivity constraints (6) are very rich and reduce to the constraints

$$x_{ij} + y_{ij} \leq y_{jj} \tag{12}$$

whenever $S = \{v_i, v_j\} \subseteq V\backslash\{v_1\}$. Indeed, when $S = \{v_i, v_j\}$ and $v_1 \notin S$, when (6) can be written as $x(\delta(i)) + x(\delta(j)) - 2x_{ij} \geq 2y_{ii} + 2y_{ij}$, and by Eqs. (4), we obtain inequalities (12). Moreover, observe that constraints (12) are stronger than the constraints

$$x_{ij} \leq 1 - y_{ij} \qquad \text{for all } v_i, v_j \in V\backslash\{v_1\},$$

and they dominate the classical inequalities:

$$x_{ij} \leq y_{ii} \text{ and } x_{ij} \leq y_{jj} \qquad \text{for all } v_i, v_j \in V\backslash\{v_1\}.$$

Furthermore, new useful constraints similar to (12) are

$$x_{1i} \leq y_{ii} \qquad \text{for all } v_i \in V\backslash\{v_1\}, \tag{13}$$

which force vertex v_i to be in the cycle if edge $[v_1, v_i]$ is included in the cycle.

Inequalities valid for the cycle polytope (Bauer [2]) are also clearly valid for the RSP. We restrict our attention to the following *2-matching inequalities:*

$$x(E(H)) + x(T) \leq \sum_{v_i \in H} y_{ii} + \frac{|T| - 1}{2} \tag{14}$$

for all $H \subset V$ and $T \subset \delta(H)$, satisfying

(i) $\{v_i, v_j\} \cap \{v_k, v_\ell\} = \emptyset$ for $[v_i, v_j], [v_k, v_\ell] \in T$ and $[v_i, v_j] \neq [v_k, v_\ell]$,
(ii) $|T| \geq 3$ and odd.

These inequalities are standard for cycle problems and their validity can be proved by adding equalities (4) for all $v_i \in H$ and inequalities $x_{ij} \leq 1$ for all $[v_i, v_j] \in T$, dividing by two, and rounding down, first, the left-hand side and, second, the right-hand side of the inequality. Constraints (14) can be rewritten in the form

$$x(\delta(H)\backslash T) + (|T| - x(T)) \geq 1,$$

which coincides with the *blossom inequalities* of the TSP polytope. This implies that when $x(T) = |T|$ then $x(\delta(H)\backslash T) \geq 1$, while when $x(T) \leq |T| - 1$, then $x(\delta(H)\backslash T) \geq 0$. Therefore, intuitively, constraints (14) impose that at least an edge in $\delta(H)\backslash T$ must be in the cycle when all the edges in T are included in the cycle.

An additional family of valid inequalities follows from the fact that some assignments are incompatible. More specifically, variables y_{ij} and y_{ik} are incompatible when $j \neq k$, and, thus, the *Stable Set Problem* (SSP) associated to those incompatibility relations between the assignment variables is a relaxation of the RSP. A similar observation was made by Avella and Sassano [1] for the p-Median Problem. This property leads (for example) to the following odd-hole (or clique) inequalities:

$$y_{ij} + y_{jk} + y_{ki} \leq 1 \tag{15}$$

for three different vertices $v_i, v_j, v_k \in V\backslash\{v_1\}$. See Padberg [15] for a partial polyhedral description of the stable set polytope.

Finally, the combination of the two mentioned relaxations (i.e., the cycle and the SSPs) produces new valid inequalities for the RSP, such as

$$x(\delta(S)) \geq 2(y_{ij} + y_{jk} + y_{ki}) \tag{16}$$

for three different vertices $v_i, v_j, v_k \in V\backslash\{v_1\}$ and all $S \subseteq V\backslash\{v_1\}$ such that $v_i, v_j, v_k \in S$.

In next section, we study conditions under which these inequalities are facet-defining for the RSP polytope. As a consequence, all the families contain facet-defining inequalities; hence, they all contribute to the polyhedral analysis of the polytope. Moreover, in our computational experiments on Euclidean instances from the literature (see Section 5), they were all relevant, except for constraints (15) and (16), which could be useful for non-Euclidean instances.

3. POLYHEDRAL ANALYSIS

We now derive some polyhedral results for model (3)–(11). Denote by P the polytope defined by the convex hull of feasible solutions to RSP and by Q the associated TSP polytope, that is,

$$P := \text{conv}\{(x, y) \in \mathbb{R}^{|E|+|A|} : (x, y) \text{ satisfies (4)–(11)}\},$$

$$Q := \{(x, y) \in P : y_{ii} = 1 \qquad \text{for all } v_i \in V\backslash\{v_1\}\}.$$

Define F as the set of free vertices, that is, a set of vertices not restricted to be in the cycle. Polytopes P and Q are linked by the intermediate polytopes:

$$P(F) := \{(x, y) \in P : y_{ii} = 1 \qquad \text{for all } v_i \notin F \cup \{v_1\}\},$$

for all $F \subseteq V\backslash\{v_1\}$, in the sense that $P(\emptyset) = Q$ and $P(V\backslash\{v_1\}) = P$. This connection allows us to extend known results from Q to P. For any $\alpha \in \mathbb{R}^{|E|}$, $\beta \in \mathbb{R}^{|A|}$, and $\gamma \in \mathbb{R}$, define the hyperplane

$$\mathcal{H}(\alpha, \beta, \gamma) := \{(x, y) \in \mathbb{R}^{|E|+|A|} : \alpha x + \beta y = \gamma\}.$$

Lemma 3.1. *Let $v_{i_1}, v_{i_2}, \ldots, v_{i_{n-1}}$ be an ordering of the vertices of $V\backslash\{v_1\}$ and let $F_k = \{v_{i_1}, \ldots, v_{i_k}\}$ for all $k = 1, \ldots, n-1$. If for each $k = 1, \ldots, n-1$ and each $v_{i_t} \in V\backslash\{v_{i_k}\}$ there exists a feasible solution (x, y) to the RSP such that* (*iii*) (*i*) $y_{i_l i_l} = 1$ *for all* $l > k$ (*i.e., all vertices of* $V \backslash F_k$ *belong to the cycle*),

(*ii*) $y_{i_l i_l} + y_{i_l 1} = 1$ *for all* $l < k$ (*i.e., all vertices of* $F_k \backslash \{v_{i_k}\}$ *are either on the cycle or assigned to the root* v_1),

(*iii*) $y_{i_k i_t} = 1$ (*i.e., vertex* v_{i_k} *does not belong to the cycle and is assigned to vertex* v_{i_t}), *and*

(*iv*) $\alpha x + \beta y = \gamma$,

then $\dim(\mathcal{H}(\alpha, \beta, \gamma) \cap P) \geq \dim(\mathcal{H}(\alpha, \beta, \gamma) \cap Q) + (|V| - 1)^2$.

Proof. We will prove by induction on $|F_k|$ that $\dim(\mathcal{H}(\alpha, \beta, \gamma) \cap P(F_k)) \geq \dim(\mathcal{H}(\alpha, \beta, \gamma) \cap Q) + |F_k|(|V| - 1)$. If $k = 0$, that is, if $F_k = \emptyset$, then this inequality holds since $P(F_k) = Q$. Now, by the induction hypothesis, there exist $\dim(\mathcal{H}(\alpha, \beta, \gamma) \cap Q) + |F_{k-1}|(|V| - 1) + 1$ affinely independent points such that $y_{j_k j_k} = 1$ in $\mathcal{H}(\alpha, \beta, \gamma) \cap P(F_{k-1})$ and, thus, in $\mathcal{H}(\alpha, \beta, \gamma) \cap P(F_k)$. An additional set of $|V| - 1$ affinely independent points such that $y_{j_k j_k} = 0$ in $\mathcal{H}(\alpha, \beta, \gamma) \cap P(F_k)$ are obtained as in the lemma's hypothesis. ■

Roughly speaking, conditions (i), (ii), and (iii) of Lemma 3.1 describe features of a large collection of RSP solutions that guarantee that they are linearly independent. These features are accomplished by tight RSP solutions for several valid inequalities, as is shown in the following propositions regarding facet-defining results:

Proposition 3.1. $\dim(P) = |E| + |V|^2 - 3|V| + 1$.

Proof. Recall that we are assuming that $|V| \geq 5$. Then,

$$\begin{aligned}
\dim(P) &= \dim(\mathcal{H}(0, 0, 0) \cap P) \\
&\geq \dim(\mathcal{H}(0, 0, 0) \cap Q) + (|V| - 1)^2 \\
&\qquad \text{(by Lemma 3.1 using any sequence)} \\
&= \dim(Q) + (|V| - 1)^2 \\
&= |E| - |V| + (|V| - 1)^2 \\
&\qquad \text{(see Grötschel and Padberg [9]).}
\end{aligned}$$

Now, $\dim(P) \leq |E| + |V|^2 - 3|V| + 1$ since the RSP can be described with $|E| + |V|^2$ variables and $|V|$ type (4), $|V| - 1$ type (5), and $|V|$ type (9)–(10) linearly independent equality constraints. ■

Another important consequence of Lemma 3.1 is that any facet-defining inequality $\alpha x + \beta y \leq \gamma$ for Q satisfying (i)–(iv) is also facet-defining for P. We will use this result for the polyhedral analysis of P.

A first consequence of Lemma 3.1 is that $x_{ij} \geq 0$ defines a facet of P for each $[v_i, v_j] \in E$, and $y_{ij} \geq 0$ defines a facet of P for each $(v_i, v_j) \in A$, $i \neq j$ and $i \neq 1$ (see Rodríguez Martín [17] for detailed proofs). Note at this stage that the inequality $x_{ij} \leq 1$ is not facet-inducing since it is dominated by the connectivity constraint (12). If $i = j$, the inequality $y_{ij} \geq 0$ is not facet-inducing since it is implied by equalities (4) and $x_{ij} \geq 0$ for $(v_i, v_j) \in F$. If $i \neq j$ and $i = 1$, then $y_{ij} = 0$ since $y_{11} = 1$, and, therefore, $y_{ij} \geq 0$ is not facet-inducing. The inequality $y_{ij} \leq 1$ is not facet-inducing since it is implied by constraints (5) and (8).

Proposition 3.2. *The valid inequality* (6) *defines a facet of* P *for each* $S \subsetneq V \backslash \{v_1\}$, $2 \leq |S| \leq |V| - 3$, $v_i \in S$.

Proof. The hyperplane defined by $x(\delta(S)) = 2 \sum_{v_j \in S} y_{ij}$ satisfies the assumptions of Lemma 3.1 for any sequence containing v_i in last position. Indeed, for any vertex $v_k \in V \backslash \{v_1, v_i\}$, there exists a simple cycle spanning $V \backslash \{v_k\}$ and with two edges in $\delta(S)$ such that, together with the assignment of v_k to each vertex of the cycle, yields (x, y) vectors of P satisfying $x(\delta(S)) = 2 \sum_{v_j \in S} y_{ij} (=2)$. Furthermore, for v_i, the last vertex of the sequence, a simple cycle spanning $(S \cup \{v_1\} \backslash \{v_i\})$ with two edges in $\delta(S)$ can be constructed. The assignment of v_i to each vertex of $S \backslash \{v_i\}$ yields $|S| - 1$ solutions (x, y) satisfying the hypotheses (i)–(iv) of Lemma 3.1. An additional $|V \backslash S|$ such solutions (x, y) are obtained by constructing a simple cycle spanning $V \backslash S$ (which contains at least three vertices) and successively assigning v_i to each vertex of $V \backslash S$. The result follows by noting that in Q the inequality $x(\delta(S)) \geq 2 \sum_{v_j \in S} y_{ij}$ reduces to $x(\delta(S)) \geq 2$, which is facet-defining (Grötschel and Padberg [9]). ■

Proposition 3.2 does not hold if $|S| = 1$ or $|S| = |V| - 2$ since constraints (6) are then dominated by (4) or $x(\delta(S)) \geq 2$, respectively. The validity of the latter inequality comes from the fact that the cycle must contain at least one vertex from S whenever $S \subset V \backslash \{v_1\}$ and $|S| \geq |V| - 2$. Proposition 3.2 is also invalid if $v_i \notin S$, since, then,

$$\begin{aligned}
x(\delta(S \cup \{v_i\})) &= x(\delta(S)) + x(\delta(i)) - 2 \sum_{v_j \in S} x_{ij} \\
&\geq 2 \sum_{v_j \in S \cup \{v_i\}} y_{ij} = 2\Big(y_{ii} + \sum_{v_j \in S} y_{ij}\Big).
\end{aligned}$$

This implies that $x(\delta(S)) \geq 2 \sum_{v_j \in S} (y_{ij} + x_{ij})$, which dominates the corresponding constraint (6).

Proposition 3.3. *The inequality* $x_{1j} \leq y_{jj}$ *defines a facet of* P *for* $j \neq 1$.

Proof. The hyperplane defined by $x_{1j} = y_{jj}$ satisfies the assumptions of Lemma 3.1. This can be seen by taking a simple cycle spanning $V \backslash \{v_k\}$ and successively assigning v_k to each vertex of the cycle. Furthermore, this cycle must contain edge $[v_1, v_j]$ whenever $v_k \neq v_j$. The result follows from the fact that $x_{1j} \leq y_{jj} = 1$ induces a facet of Q (Grötschel and Padberg [9]). ■

The constraints $x_{i1} + y_{i1} \leq 1$ for all $i \neq 1$ do not induce facets of P since these are dominated by the constraints $x_{1i} \leq y_{ii}$. Indeed, $x_{1i} \leq y_{ii} = 1 - \sum_{v_j \in \setminus\{v_i\}} y_{ij} \leq 1 - y_{i1}$.

Proposition 3.4. *The inequality* (14) *for each $H \subset V$ and $T \subset \delta(H)$ satisfying*

(i) $\{v_i, v_j\} \cap \{v_k, v_\ell\} = \varnothing$ for $[v_i, v_j], [v_k, v_\ell] \in T$ and $[v_i, v_j] \neq [v_k, v_\ell]$,
(ii) $|T| \geq 3$ and odd

defines a facet of P when $|V| \geq 6$.

Proof. The hyperplane defined by $x(E(H)) + x(T) \leq \sum_{v_i \in H} y_{ii} + (|T| - 1)/2$ satisfies the assumptions of Lemma 3.1. Indeed, for each vertex $v_k \in V\backslash\{v_1\}$, it is trivial to design a simple cycle spanning $V\backslash\{v_k\}$ and satisfying hypotheses (i)–(iv) of Lemma 3.1, and v_k can be assigned to each vertex of the cycle. The result follows from the fact that $x(E(H)) + x(T) \leq \sum_{v_i \in H} y_{ii} + (|T| - 1)/2$ defines a facet of Q (Grötschel and Padberg [9]). ■

We now turn to the two inequalities (15) and (16) related to the SSP relaxation. The proofs of the following two results are provided in the Appendix:

Proposition 3.5. *The inequality* (15) *defines a facet of P for each three different vertices $v_i, v_j, v_k \in V\backslash\{v_1\}$.*

If one vertex, say v_k, coincides with the root v_1, Proposition 3.5 does not hold, since, then, (15) is dominated by (12). Indeed, in this case,

$$y_{ij} + y_{j1} + y_{1i} = y_{ij} + y_{j1} \leq y_{jj} + y_{j1} \leq \sum_{v_l \in V} y_{jl} = 1.$$

Proposition 3.6. *The inequality* (16) *defines a facet of P when $|V| \geq 7$ for each three different vertices $v_i, v_j, v_k \in V\backslash\{v_1\}$ and each $S \subseteq V\backslash\{v_1\}$ such that $v_i, v_j, v_k \in S$ and $4 \leq |S| \leq |V| - 3$.*

Observe that inequalities (16) are not facet-defining when $|S| = 3$, since, in that case, they are the sum of inequalities (12) for arcs (v_i, v_j), (v_j, v_k), and (v_k, v_i). Also, (16) cannot be facet-defining if $|V\backslash S| \leq 2$, since, then, $x(\delta(S))$ must be equal to at least two.

4. BRANCH-AND-CUT ALGORITHM

We now outline the main ingredients of our branch-and-cut algorithm for finding an optimal solution cycle $\bar{C}$ of the RSP. Let $V(\bar{C})$ be the set of vertices in the solution.

STEP 1 (Initialization). Set the iteration count $t := 1$. Compute an upper bound $\bar{z}$ on the optimal RSP as follows: Starting with $V(\bar{C}) := \{v_1\}$, successively insert a vertex $v_i \notin V(\bar{C})$ to minimize

$$L(i, \lambda) := \lambda\, inc(\bar{C}, i) + (1 - \lambda)(-dec(\bar{C}, i)),$$

where $inc(\bar{C}, i)$ is the minimum increment produced in the ring cost of cycle $\bar{C}$ when inserting v_i, and $dec(\bar{C}, i)$ is the decrease produced in the assignment cost. This operation is repeated as long as $L(i, \lambda)$ decreases with a new insertion. This is done for each $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and the best solution is selected. The values of λ were chosen so as to generate five rings, each with a different tradeoff between the ring cost and the assignment cost.

On the other hand, the initial linear subproblem is then defined as

$$\text{minimize} \sum_{[v_i,v_j] \in E} c_{ij} x_{ij} + \sum_{(v_i,v_j) \in A} d_{ij} y_{ij},$$

subject to

$$x(\delta(i)) = 2y_{ii} \qquad \text{for all } v_i \in V$$

$$\sum_{v_j \in V} y_{ij} = 1 \qquad \text{for all } v_i \in V\backslash\{v_1\}$$

$$y_{11} = 1$$

$$x_{1i} \leq y_{ii} \qquad \text{for all } v_i \in V\backslash\{v_1\},$$

and the subproblem is solved and inserted in a list $\mathcal{L}$.

STEP 2 (Termination check and subproblem selection). If the list $\mathcal{L}$ is empty, stop. Otherwise, select a subproblem from the list according to a best-first policy, that is, select the subproblem having lowest objective value.

STEP 3 (Subproblem solution). Set $t := t + 1$. Let z be the solution objective value. If $z \geq \bar{z}$, go to Step 2. Otherwise, if the solution is feasible for the RSP, set $\bar{z} := z$, update $\bar{C}$, and go to Step 2.

STEP 4 (LP-based heuristic). If the solution is fractional and t is a multiple of 5, apply the following heuristic: Given the fractional solution (x^*, y^*), sort the x^*_{ij} variables in decreasing order of their values and take, in turn, their associated edges until a cycle C^* is obtained. If v_1 is not in C^*, introduce it in the best position. Finally, apply the classical 2-optimallity procedure to further improve the ring cost of C^*. Let z^* be the objective function value of solution C^*. If $z^* \leq \bar{z}$, set $\bar{z} := z^*$ and update $\bar{C}$ with C^*.

STEP 5 (Constraint separation and generation). Introduce up to 300 violated connectivity constraints (6), 2-matching constraints (14), and constraints (15) and (16). If no constraints can be generated, go to Step 6. Otherwise, go to Step 3.

STEP 6 (Branching). Create two subproblems by branching on a fractional y_{ii} or x_{ij} variable. The first branching strategy consists of finding a y_{ii} variable. To this end, we applied the "strong branching" rule (see Jünger and Thienel [10] for details) within the five y_{ii} variables with fractional value closest to 0.5. If all these variables are integer, select a x_{ij} variable using the same criterion. Insert both subproblems in $\mathcal{L}$ and go to Step 2.

The initial heuristic is a quite simple and intuitive method, since no advantage was found in using a more sophisticated approach.

We have developed the following separation procedures for Step 5. Denote by $G^* = (V^*, E^*)$ the *support graph* associated with a given (fractional) solution (x^*, y^*), that is, $V^* := \{v_i \in V : 0 < y^*_{ii} < 1\}$ and $E^* := \{[v_i, v_j] \in E : 0 < x^*_{ij} < 1\}$. We also define $A^* := \{(v_i, v_j) \in A : i \neq j, 0 < y^*_{ij} < 1\}$.

Separation of Connectivity Constraints (6)

The inequalities (6) involving sets S such that $|S| = 2$ reduce to (12), and any of them can be violated only if the corresponding edge $[v_i, v_j]$ belongs to E^* or the corresponding arc (v_i, v_j) belongs to A^*. So, they are separated by examining only edges in E^* and arcs in A^*. Next, if the support graph G^* is not connected, then each subset $S \subset V$ corresponding to a connected component and such that $v_1 \notin S$ generates a violated connectivity constraint for each $v_i \notin S$. If G^* is connected, finding a most violated connectivity constraint $x(\delta(S)) \geq 2 \sum_{v_j \in S} y_{ij}$ is equivalent to finding the largest violation of $x(\delta(S)) + 2 \sum_{v_j \notin S} y_{ij} \geq 2$. For a given vertex $v_i \in V \backslash \{v_1\}$, this reduces to a maximum-flow problem defined as follows: Let $G^{**} = (V^{**}, E^{**})$ be a graph such that $V^{**} = V^* \cap \{v_{n+1}\}$ and $E^{**} = E^* \cup \{[v_{n+1}, v_j] : v_j \in V^*\}$. The capacity of each edge $e \in E^*$ is equal to x^*_e and that of the new edges $[v_{n+1}, v_j]$ is equal to $2y^*_{ij}$. Let $S' \subset V^{**}$ be such that $v_{n+1} \in S'$, $v_1 \notin S'$ and the capacity Δ of the cut $\delta(S')$ in G^{**} is minimum. If $\Delta \geq 2$, there is no connectivity constraint involving v_i violated by the current solution (x^*, y^*). Otherwise, $S = S' \backslash \{v_{n+1}\}$ defines a most violated connectivity constraint (6) involving v_i.

Note that constraints similar to (6) were introduced by Lee et al. [13] for the Steiner RSP. However, these authors only describe a separation procedure for the subset of weaker constraints:

$$x(\delta(S)) \geq 2(y_{ii} + y_{jj} - 1) \quad \text{for all } S \subset V : v_i \in S,\ v_j \notin S. \tag{17}$$

Separation of 2-Matching Constraints (14)

To separate the 2-matching inequalities, we use the heuristic procedure proposed in Fischetti et al. [6]. This algorithm can identify several violated 2-matching constraints for the current fractional solution. Briefly, it consists of considering each connected component H of G^* as the handle of a possibly violated generalized 2-matching inequality whose 2-node teeth correspond to the edges $e \in \delta(H)$ with $x^*_e = 1$ (if the number of these edges is even, the inequality is clearly rejected).

Separation of Constraints (15) and (16)

Clearly, inequalities (15) can be separated through a complete enumeration of v_i, v_j, v_k such that $y^*_{ij} > 0$, $y^*_{jk} > 0$, and $y^*_{ki} > 0$. Indeed, when (say) $y_{ij} = 0$ and (12) holds, then (15) also holds. In a similar way, for each v_i, v_j, v_k such that $y^*_{ij} > 0$, $y^*_{jk} > 0$ and $y^*_{ki} > 0$, a min-cut separating v_1 from $\{v_i, v_j, v_k\}$ in G^* gives the most-violated constraint (16), if any. Again, when (say) $y_{ij} = 0$ and (6) holds, then (16) also holds.

5. COMPUTATIONAL RESULTS

The branch-and-cut algorithm described in the previous sections was implemented in the C++ programming language. ABACUS 2.3 linked with CPLEX 6.0 was used as a branch-and-cut framework. See Jünger and Thienel [10] for details on this software. The performance of the algorithm was tested on three different classes of test instances. The root was always chosen as the first vertex.

Class I is based on TSP instances from TSPLIB 2.1 (Reinelt [16]) involving between 50 and 200 vertices (problems *eil51* to *kroB200*). Denote by l_{ij} the distance between vertices v_i and v_j given in the TSP files. To obtain optimal solutions visiting approximately 100, 75, 50, and 25% of the total number of vertices in the instances, we set $c_{ij} = \lceil \alpha l_{ij} \rceil$, $d_{ij} = \lceil (10 - \alpha) l_{ij} \rceil$ for $\alpha \in \{3, 5, 7, 9\}$, and $d_{ii} = 0$ for all $v_i \in V$.

Class II was generated in the following way: We generated vertices with coordinates in [0, 1000] × [0, 1000] and computed l_{ij} as the Euclidean distance between v_i and v_j rounded up to the nearest integer. Costs c_{ij}, d_{ij}, and d_{ii} are defined as in Class I, using the same values of α. For each pair $(|V|, \alpha)$, we generated 10 instances.

Class III consists of instances generated as in Lee et al. [13]. We first generated vertices with coordinates in [0, 1000] × [0, 1000] and then computed l_{ij} as the Euclidean distance between v_i and v_j rounded up to the nearest integer. Costs $c_{ij} := d_{ij} := l_{ij}$ and costs d_{ii} were randomly generated in the interval [0, 1000] for all $v_i \in V$. For each value $|V|$, we generated 10 instances.

Tables 1–4 summarize the computational behavior of our branch-and-cut code on the three classes of instances. The column headings are defined as follows:

Name: problem name (for Class I);
$|V|$: number of vertices (for Classes II and III);
α: scale factor described above (for Classes I and II);
succ: number of instances solved to optimallity over 10 trials (for Classes II and III);
p^*: percentage of vertices visited by the optimal cycle;
opt: optimal objective value (for Class I);

TABLE 1. Computational results for instances in Class I.

Name	α	%-UB0	h-Time	p^*	opt	%-LB	%-UB	Pair	Sec	2mat	Nodes	Time
eil51	3	107.51	0:00:00	100.00	1278	100.00	107.51	4	155	91	3	0:00:02
eil51	5	102.76	0:00:00	74.51	1995	100.00	100.00	61	42	0	1	0:00:01
eil51	7	101.47	0:00:00	33.33	2113	100.00	100.00	238	1251	0	1	0:00:10
eil51	9	101.29	0:00:00	11.76	1244	100.00	100.00	529	883	0	1	0:00:10
berlin52	3	105.06	0:00:00	100.00	22,626	100.00	100.00	12	46	0	1	0:00:01
berlin52	5	102.78	0:00:00	78.85	36,115	100.00	100.00	85	957	36	1	0:00:04
berlin52	7	100.94	0:00:00	46.15	37,376	100.00	100.00	221	655	1	1	0:00:06
berlin52	9	100.78	0:00:00	9.62	20,361	100.00	100.00	539	941	0	1	0:00:12
brazil58	3	102.16	0:00:01	100.00	76,185	100.00	100.00	18	105	3	1	0:00:02
brazil58	5	101.46	0:00:01	68.97	115,045	100.00	100.00	104	2994	285	1	0:00:14
brazil58	7	100.24	0:00:00	48.28	126,807	100.00	100.00	237	1543	2	1	0:00:12
brazil58	9	100.00	0:00:00	15.52	83,690	100.00	100.00	577	1502	0	1	0:00:28
st70	3	103.41	0:00:01	100.00	2025	99.80	103.41	32	660	110	11	0:00:13
st70	5	102.41	0:00:01	78.57	3110	99.84	102.41	110	2837	243	3	0:00:24
st70	7	101.41	0:00:01	42.86	3402	100.00	100.00	283	1920	0	1	0:00:20
st70	9	100.15	0:00:00	25.71	2610	100.00	100.00	806	1935	0	1	0:01:10
eil76	3	107.25	0:00:02	100.00	1614	100.00	100.00	6	139	43	1	0:00:05
eil76	5	100.61	0:00:02	73.68	2460	99.92	100.61	113	1858	56	5	0:00:21
eil76	7	102.76	0:00:01	42.11	2504	100.00	100.00	343	2228	0	1	0:00:37
eil76	9	101.35	0:00:00	15.79	1710	100.00	100.00	930	2440	0	1	0:02:28
pr76	3	101.55	0:00:02	100.00	324,477	98.59	101.55	128	15270	6043	727	0:17:39
pr76	5	106.21	0:00:02	80.26	500,395	99.66	106.21	171	8847	1489	71	0:02:27
pr76	7	103.89	0:00:01	53.95	555,858	99.80	103.89	350	6744	254	13	0:01:54
pr76	9	100.72	0:00:00	17.11	424,359	100.00	100.00	874	3011	0	1	0:03:04
gr96	3	102.75	0:00:15	100.00	163,935	99.30	102.75	38	1608	415	31	0:01:43
gr96	5	104.31	0:00:13	79.17	252,850	100.00	100.00	144	4802	292	1	0:01:06
gr96	7	105.97	0:00:07	50.00	275,599	100.00	100.00	409	3254	2	1	0:01:08
gr96	9	105.99	0:00:02	27.08	232,823	100.00	100.00	1236	3766	0	1	0:06:53
rat99	3	107.35	0:00:06	100.00	3633	99.89	107.35	12	1178	234	13	0:00:38
rat99	5	103.65	0:00:05	89.90	5885	100.00	100.00	150	2437	300	1	0:00:35
rat99	7	104.26	0:00:02	42.42	6436	100.00	100.00	480	3197	0	1	0:01:20
rat99	9	105.01	0:00:01	21.21	5150	100.00	100.00	1285	4541	0	1	0:07:19
kroA100	3	100.86	0:00:06	100.00	63,846	99.80	100.86	24	886	211	11	0:00:37
kroA100	5	100.21	0:00:05	80.00	100,785	99.78	100.21	149	6285	246	3	0:01:26
kroA100	7	106.18	0:00:03	55.00	115,388	100.00	100.00	428	3368	4	1	0:01:16
kroA100	9	103.15	0:00:00	21.00	94,265	100.00	100.00	1250	3150	0	1	0:05:57
kroB100	3	105.42	0:00:06	100.00	66,423	99.50	105.42	30	2051	310	39	0:01:38
kroB100	5	104.35	0:00:05	77.00	104,550	100.00	100.00	147	4319	271	1	0:00:57
kroB100	7	103.95	0:00:02	46.00	118,111	100.00	100.00	435	3322	0	1	0:01:17
kroB100	9	100.99	0:00:00	16.00	93,938	100.00	100.00	1303	4238	0	1	0:07:34
kroC100	3	102.42	0:00:06	100.00	62,247	99.81	102.42	48	1074	215	9	0:00:33
kroC100	5	103.43	0:00:05	81.00	99,065	100.00	100.00	141	3283	280	1	0:00:45
kroC100	7	102.40	0:00:03	50.00	113,533	100.00	100.00	438	3200	0	1	0:01:07
kroC100	9	102.07	0:00:01	23.00	92,894	100.00	100.00	1319	4056	0	1	0:07:05
kroD100	3	105.58	0:00:06	100.00	63,882	99.88	105.58	40	948	186	5	0:00:32
kroD100	5	105.25	0:00:05	82.00	101,645	100.00	100.00	154	2651	119	1	0:00:38
kroD100	7	101.07	0:00:02	47.00	116,849	99.94	101.07	440	4213	7	3	0:01:50
kroD100	9	102.82	0:00:00	23.00	92,102	100.00	100.00	1251	4123	0	1	0:06:58
kroE100	3	104.55	0:00:06	100.00	66,204	99.28	104.55	48	821	374	41	0:01:44
kroE100	5	103.67	0:00:05	77.00	104,915	99.94	103.67	170	6678	447	11	0:01:54
kroE100	7	105.60	0:00:02	51.00	116,471	100.00	100.00	441	3010	1	1	0:01:06
kroE100	9	100.96	0:00:01	20.00	96,116	100.00	100.00	1228	5068	0	1	0:08:25
rd100	3	101.67	0:00:06	100.00	23,730	100.00	100.00	28	524	122	1	0:00:16
rd100	5	102.34	0:00:05	76.00	37,975	99.87	102.34	165	6766	282	5	0:01:40
rd100	7	102.61	0:00:02	44.00	40,915	100.00	100.00	433	5848	26	1	0:01:53
rd100	9	102.59	0:00:01	21.00	31,776	100.00	100.00	1287	4368	0	1	0:07:01
eil101	3	106.84	0:00:06	100.00	1887	99.84	106.84	16	824	227	19	0:00:46
eil101	5	103.61	0:00:05	72.28	2905	100.00	100.00	301	1810	257	1	0:00:31
eil101	7	102.29	0:00:02	38.61	2926	99.93	102.29	1680	7918	18	7	0:04:00
eil101	9	101.48	0:00:00	17.82	1955	100.00	100.00	3291	4970	0	1	0:08:18
lin105	3	103.43	0:00:07	100.00	43,137	100.00	100.00	56	808	145	1	0:00:21
lin105	5	100.81	0:00:06	80.95	69,365	100.00	100.00	233	2588	341	1	0:00:43
lin105	7	107.97	0:00:04	53.33	83,597	100.00	100.00	1149	4047	6	1	0:01:38
lin105	9	102.37	0:00:02	31.43	69,920	100.00	100.00	3995	4998	0	1	0:07:30
Averages	3	104.24	0:00:05	100.00	61,695.56	99.73	103.02	33.75	1693.56	545.56	57.13	0:01:41
	5	102.99	0:00:04	78.13	96,191.25	99.94	100.97	149.88	3697.13	309.00	6.75	0:00:52
	7	103.31	0:00:02	46.50	107,242.81	99.98	100.45	500.31	3482.38	20.06	2.25	0:01:15
	9	101.98	0:00:01	19.82	84,057.06	100.00	100.00	1356.25	3374.38	0.00	1.00	0:05:02

TABLE 2. Computational results for instances in Class I (continued).

Name	α	%-UB0	h-Time	p^*	Opt	%-LB	%-UB	Pair	Sec	2mat	Nodes	Time
pr107	3	100.36	0:00:08	100.00	132,909	100.00	100.00	0	1171	15	1	0:00:23
pr107	5	100.89	0:00:06	68.22	210,465	100.00	100.00	327	3418	169	1	0:00:50
pr107	7	100.63	0:00:03	42.99	259,571	100.00	100.00	1198	4979	1	1	0:01:52
pr107	9	101.08	0:00:01	26.17	264,918	100.00	100.00	4218	5366	0	1	0:06:45
gr120	3	103.26	0:00:14	100.00	20,826	99.92	103.26	40	1337	186	0	0:01:01
gr120	5	103.75	0:00:10	75.83	31,480	100.00	100.00	420	8782	366	1	0:02:38
gr120	7	102.19	0:00:06	41.67	32,301	100.00	100.00	1211	6133	0	1	0:03:19
gr120	9	103.19	0:00:01	22.50	24,322	100.00	100.00	3799	6332	0	1	0:11:34
pr124	3	104.96	0:00:14	100.00	177,090	98.82	104.96	251	10,055	1594	195	0:12:32
pr124	5	103.45	0:00:13	90.32	286,115	99.79	103.45	559	7777	782	17	0:03:58
pr124	7	102.48	0:00:10	66.94	358,853	100.00	100.00	1263	7590	21	1	0:03:27
pr124	9	102.65	0:00:03	39.52	340,153	100.00	100.00	5122	7999	1	1	0:16:05
bier127	3	104.70	0:00:15	100.00	354,846	99.84	104.70	64	1883	341	29	0:02:27
bier127	5	108.23	0:00:13	76.38	539,955	100.00	108.23	526	10,150	638	3	0:03:32
bier127	7	101.35	0:00:06	44.09	567,110	99.99	101.35	2140	11,081	27	5	0:07:11
bier127	9	100.24	0:00:01	14.96	347,845	100.00	100.00	5989	7333	0	1	0:36:52
ch130	3	107.09	0:00:17	100.00	18,330	99.85	107.09	100	2049	384	21	0:02:07
ch130	5	106.60	0:00:15	83.08	28,790	100.00	100.00	400	7044	608	1	0:02:38
ch130	7	103.87	0:00:08	47.69	32,707	99.64	100.77	3183	19,575	143	15	0:11:24
ch130	9	101.72	0:00:01	16.15	23,639	100.00	100.00	5179	7112	0	1	0:21:56
pr136	3	104.15	0:00:20	100.00	290,316	99.45	104.15	42	11,092	5600	147	0:15:04
pr136	5	101.43	0:00:18	86.76	468,520	100.00	100.00	401	6192	437	1	0:02:23
pr136	7	102.59	0:00:06	43.38	491,981	100.00	100.00	1786	8073	0	1	0:06:01
pr136	9	104.14	0:00:02	25.74	387,327	100.00	100.00	6152	10,262	0	1	0:25:33
gr137	3	104.13	0:02:02	100.00	208,929	99.76	104.13	46	1764	348	13	0:02:36
gr137	5	101.16	0:00:57	83.94	329,465	99.74	101.16	603	20,033	1097	15	0:08:59
gr137	7	102.44	0:00:36	54.74	366,022	100.00	100.00	1965	11,516	16	1	0:07:06
gr137	9	107.02	0:00:05	26.28	335,009	100.00	100.00	6359	10,498	0	1	0:31:09
pr144	3	103.07	0:00:25	100.00	175,611	99.83	103.07	126	1830	232	21	0:02:35
pr144	5	104.87	0:00:24	94.44	290,945	99.55	104.87	1108	45,085	4218	323	0:51:01
pr144	7	103.83	0:00:18	63.89	383,041	100.00	100.00	1607	15,046	236	1	0:07:22
pr144	9	104.68	0:00:03	23.61	366,833	100.00	100.00	7966	11,221	0	1	0:25:44
ch150	3	104.44	0:00:29	100.00	19,584	99.79	104.44	77	4270	547	35	0:05:07
ch150	5	104.64	0:00:25	77.33	31,170	99.98	100.00	550	10,366	456	3	0:05:15
ch150	7	104.02	0:00:13	48.00	34,930	99.99	104.02	1891	10,208	3	3	0:09:21
ch150	9	104.50	0:00:02	18.67	26,371	100.00	100.00	7689	12,746	0	1	0:54:12
kroA150	3	103.53	0:00:29	100.00	79,572	99.49	103.53	84	11,656	2161	115	0:17:05
kroA150	5	102.62	0:00:24	80.67	125,435	100.00	100.00	399	7538	695	1	0:03:44
kroA150	7	103.97	0:00:13	48.67	140,961	99.76	103.97	2106	10,498	90	5	0:09:05
kroA150	9	102.79	0:00:03	19.33	113,080	100.00	100.00	7254	10,606	0	1	0:45:36
kroB150	3	107.31	0:00:29	100.00	78,390	99.51	107.31	78	4667	1491	65	0:10:33
kroB150	5	104.52	0:00:25	79.33	122,875	99.49	104.52	4171	181,892	6524	221	1:52:12
kroB150	7	104.48	0:00:12	50.67	135,382	100.00	100.00	1729	8595	1	1	0:07:01
kroB150	9	102.02	0:00:03	18.67	108,885	100.00	100.00	7372	10,483	0	1	0:43:22
pr152	3	103.41	0:00:31	100.00	221,046	99.51	103.41	162	5210	2296	45	0:09:42
pr152	5	100.00	0:00:34	87.50	376,155	96.05	100.00	1595	58,290	6674	285	>2:00:00
pr152	7	100.00	0:00:18	51.32	475,052	96.66	100.00	5576	84188	2042	59	>2:00:00
pr152	9	101.78	0:00:03	21.05	475,440	100.00	100.00	10,902	14,875	0	1	0:52:15
u159	3	107.83	0:00:37	100.00	126,240	99.80	107.83	58	1346	610	7	0:02:27
u159	5	104.35	0:00:33	84.28	204,250	99.97	104.35	673	16,748	897	7	0:09:08
u159	7	101.92	0:00:20	55.97	235,221	100.00	100.00	2040	10,668	3	1	0:09:24
u159	9	101.92	0:00:05	25.79	199,552	100.00	100.00	8388	14,736	0	1	0:59:50
rat195	3	108.44	0:01:23	100.00	6969	99.68	108.44	78	17,482	2048	61	0:29:41
rat195	5	108.30	0:01:05	84.62	11,320	99.92	108.30	562	10,841	1033	11	0:12:21
rat195	7	102.53	0:00:28	40.00	12,319	100.00	100.00	2826	21,795	14	1	0:34:25
rat195	9	100.00	0:00:07	16.92	9395	94.21	100.00	8965	16,115	0	1	>2:00:00
d198	3	102.50	0:01:28	100.00	47,340	99.74	102.50	60	10,573	1102	65	0:25:54
d198	5	102.49	0:01:10	79.80	76,945	99.94	102.49	2516	106,574	7424	5	1:12:20
d198	7	102.52	0:00:38	51.01	94,300	99.87	101.34	8298	79,000	716	59	1:48:59
d198	9	100.00	0:00:09	19.19	97,899	96.34	100.00	15,536	26,958	0	1	>2:00:00
kroA200	3	100.00	0:02:05	100.00	93,699	93.59	100.00	234	23,765	2930	129	>2:00:00
kroA200	5	107.78	0:01:12	74.00	138,885	99.85	107.78	2410	149,600	3181	39	>2:00:00
kroA200	7	103.77	0:00:43	49.00	158,227	99.83	102.29	7169	72,865	919	37	1:31:05
kroA200	9	100.00	0:00:09	19.00	124,678	97.15	100.00	10,387	16,377	0	1	>2:00:00
kroB200	3	106.26	0:01:32	100.00	88,311	99.81	106.26	106	4080	868	33	0:13:44
kroB200	5	104.97	0:01:18	77.00	138,905	99.93	104.97	1025	29,056	1023	7	0:25:44
kroB200	7	101.96	0:00:48	49.00	156,638	100.00	100.00	2854	16,926	4	1	0:25:19
kroB200	9	100.00	0:00:08	18.50	127,800	95.13	100.00	10,055	14,849	0	1	>2:00:00
Averages	3	104.44	0:00:42	100.00	125,882.82	99.32	104.42	94.47	6719.41	1338.41	58.29	0:09:00
	5	104.12	0:00:34	81.38	200,686.76	99.66	102.95	1073.24	39,963.88	2130.71	55.35	0:18:38
	7	102.62	0:00:19	49.94	231,448.00	99.75	100.81	2873.06	23,455.06	249.18	11.35	0:20:08
	9	102.22	0:00:04	21.89	198,420.35	98.99	100.00	7725.41	11,933.41	0.06	1.00	0:25:21

TABLE 3. Computational results for instances in Class II.

$|V|$	α	Succ	p^*	%-UB0	h-Time	%-LB	%-UB	Pair	Sec	2mat	Nodes	Time
50	3	10	100.00	103.06	0:00:00	99.59	102.30	16.60	437.70	94.10	12.60	0:00:05
	5	10	78.60	100.94	0:00:00	99.91	100.11	70.70	941.30	69.10	2.00	0:00:04
	7	10	50.40	102.53	0:00:00	100.00	100.00	201.50	971.30	1.90	1.00	0:00:06
	9	10	12.40	100.95	0:00:00	100.00	100.00	543.50	977.90	0.00	1.00	0:00:15
75	3	10	100.00	104.30	0:00:02	99.83	102.70	22.20	548.70	148.40	9.20	0:00:15
	5	10	78.67	103.40	0:00:02	99.90	100.74	113.50	2777.90	207.60	7.00	0:00:32
	7	10	49.33	102.46	0:00:01	99.99	100.28	321.60	2364.80	5.00	1.40	0:00:33
	9	10	15.87	101.47	0:00:00	100.00	100.00	895.60	2279.60	0.00	1.00	0:02:02
100	3	10	100.00	104.40	0:00:06	99.74	104.08	35.30	1520.10	398.70	19.40	0:01:07
	5	10	77.70	103.96	0:00:05	99.87	102.08	159.90	8149.40	652.20	19.40	0:02:49
	7	10	46.00	102.35	0:00:02	99.78	100.89	514.60	31,894.50	988.70	74.40	0:12:53
	9	10	19.40	102.31	0:00:01	100.00	100.00	1262.40	4051.60	0.00	1.00	0:07:36
125	3	10	100.00	106.17	0:00:14	99.77	105.31	51.00	2949.20	481.30	24.00	0:02:44
	5	10	80.08	104.92	0:00:12	99.92	103.48	520.40	13,255.80	740.40	10.00	0:05:02
	7	10	45.44	103.09	0:00:06	99.87	100.51	1902.60	10,757.70	140.50	6.40	0:06:23
	9	10	18.16	102.36	0:00:01	100.00	100.00	5409.70	7471.00	0.00	1.00	0:22:05
150	3	10	100.00	105.66	0:00:29	99.75	104.95	118.20	7356.40	1350.30	72.00	0:10:37
	5	10	80.20	105.07	0:00:25	99.91	102.41	775.90	32,248.70	1494.70	21.00	0:16:30
	7	10	46.67	103.57	0:00:12	99.91	101.38	3718.50	23,575.00	415.70	32.20	0:23:09
	9	10	18.47	102.09	0:00:02	100.00	100.00	7424.30	11,163.00	0.00	1.00	0:54:17
175	3	10	100.00	106.69	0:00:54	99.84	105.96	94.90	4248.90	858.80	29.60	0:08:10
	5	8	79.71	105.12	0:00:46	99.88	103.20	1128.38	39,025.38	1897.50	17.50	0:27:03
	7	9	46.16	103.53	0:00:23	99.89	101.90	4206.56	30,322.00	478.78	19.00	0:34:58
	9	9	18.16	102.98	0:00:04	100.00	100.00	9630.22	15,011.78	0.11	1.00	1:43:30
200	3	10	100.00	106.74	0:01:32	99.78	106.74	169.60	12,025.50	1937.70	84.80	0:32:20
	5	9	79.56	104.57	0:01:19	99.93	103.53	1006.22	40,507.22	1871.78	10.33	0:33:14
	7	8	47.75	104.17	0:00:38	99.92	102.45	4383.25	32,413.25	369.13	14.00	0:48:01
	9	0										
Averages	3	10.00	100.00	105.29	0:00:28	99.76	104.58	72.54	4155.21	752.76	35.94	0:07:54
	5	9.57	79.22	104.00	0:00:24	99.90	102.22	539.29	19,557.96	990.47	12.46	0:12:11
	7	9.57	47.39	103.10	0:00:12	99.91	101.06	2178.37	18,899.79	342.82	21.20	0:18:00
	9	9.83	17.08	102.03	0:00:01	100.00	100.00	4194.29	6825.81	0.02	1.00	0:31:38

%-UB0: percentage ratio UB/opt, where UB is the objective function value of the initial heuristic solution (i.e., the value of the heuristic solution computed in Step 1 of the branch-and-cut algorithm);

h-time: initial heuristic time;

%-LB: percentage LB/opt, where LB is the objective value of the LP-relaxation computed at the root node of the branch-decision tree (i.e., the value of the fractional so-

TABLE 4. Computational results for instances in Class III.

	With 2-matching constraints (14)											Without	
$|V|$	Succ	p^*	%-UB0	h-Time	%-LB	%-UB	Pair	Sec	2mat	Nodes	Time	%-LB	%-UB
10	10	36.00	101.12	0:00:00	100.00	100.00	12.20	12.50	0.00	1.00	0:00:00	100.00	100.00
20	10	25.00	101.74	0:00:00	100.00	100.00	61.90	92.70	0.00	1.00	0:00:00	100.00	100.00
30	10	25.33	103.29	0:00:00	100.00	100.00	104.20	133.50	0.50	1.00	0:00:00	100.00	100.00
40	10	23.00	101.13	0:00:00	100.00	100.00	154.00	258.10	1.70	1.00	0:00:01	100.00	100.00
50	10	20.80	103.74	0:00:00	100.00	100.00	225.90	338.10	1.50	1.00	0:00:02	100.00	100.00
75	10	20.53	101.87	0:00:00	99.99	100.03	401.20	1008.60	5.20	1.20	0:00:12	99.96	100.03
100	10	19.50	103.20	0:00:01	99.98	100.17	561.60	1777.40	14.30	1.60	0:00:35	99.96	100.17
125	10	18.08	102.55	0:00:01	100.00	100.18	1013.00	2882.60	22.10	1.40	0:01:16	99.95	100.50
150	10	17.20	102.37	0:00:03	99.98	100.44	1348.10	4785.40	25.10	2.00	0:02:53	99.95	100.58
175	10	17.01	103.29	0:00:05	99.96	100.65	2534.10	13,865.80	98.00	6.20	0:12:33	99.84	102.26
200	10	16.50	102.32	0:00:08	99.96	100.79	3177.60	26,594.00	165.40	11.00	0:28:22	99.86	101.62
250	8	15.80	103.39	0:00:17	99.98	101.32	3792.75	22,781.75	133.63	4.25	0:32:54	99.90	102.99
300	4	14.75	102.62	0:00:30	99.99	100.25	9220.25	40,934.25	191.25	1.50	1:08:41	99.91	102.13
Averages	9.38	20.73	102.51	0:00:05	99.99	100.29	1738.98	8881.90	50.67	2.63	0:11:21	99.95	100.79

lutiion of the LP-relaxation when the algorithm first goes to Step 6);

%-UB: percentage ratio UB/opt, where UB is the objective value of the best solution at the root node of the branch-decision tree;

pair: number of Constraints (12) generaged;

sec: number of Constraints (6) with $|S| > 2$ generated;

2mat: number of Constraints (14) generated;

nodes: total number of nodes examined (1 means that the problem required no branching);

time: total computing time.

The computing times reported are expressed in the format *h:mm:ss* and refer to CPU time on a Pentium III personal computer running at 866 MHz. We imposed a time limit of 2 h for each run. For the instances exceeding the time limit, we report ">2:00:00" in the *time* column and compute the corresponding results by considering the best available solution as optimal. Hence, for the time-limit instances, the column %-LB gives an overestimation of the percentage approximation error. Tables 1 and 2 refer to Class I (TSPLIB-based instances). Tables 3 and 4 refer to instances in Class II and Class III, respectively, and contain average results over 10 trials. Last lines provide average results overall the trials for each table.

Preliminary experiments revealed that inequalities (15) and (16) are not useful for solving these benchmark instances in the sense that they were not violated after all others had been separated. Moreover, even when violations were detected at the root node, introducing these constraints did not help to reduce the gap between the lower and the upper bounds and did not reduce the computing time. Hence, we decided not to use these inequalities in our tests. Nevertheless, one could expect that these inequalities should be useful for solving some non-Euclidean distance RSP instances.

The results presented in Tables 1–3 indicate that the proposed branch-and-cut algorithm can solve instances involving up to 200 vertices within modest computing times. For a given problem size, the most difficult instances tend to be those where relatively few vertices belong to an optimal cycle. This can be explained by the fact that the location component of the problem is then important. At the other extreme, instances in which all vertices belong to an optimal cycle reduce to a TSP which is a relatively easy problem for the values of $|V|$ considered. The %-LB column indicates that the lower bound developed at the root of the branch-and-cut tree is very tight, typically within 0.5% of the optimum. The heuristic is also very powerful and usually yields a solution within 5% of the optimum. It seems to perform better when not all vertices belong to the cycle. All valid inequalities, apart from (15) and (16), are frequently generated within the search tree. The number of nodes in the tree is relatively low, and several instances (in particular, the TSPLIB-based instances) are solved without any branching. Overall, we were able to solve to optimality 124 of the 132 TSPLIB based instances in Class I. For the random instances in Class II, the corresponding statistic is 267 of 280, and the largest instances involved 200 vertices. As observed, the last lines show the average performance of the algorithm. In particular, the gap of the LP-relaxation and of the heuristic approach at the end of the root node are very close to 0.2 and 2%, respectively. The average time to solve a random instance with up to 200 vertices is under 0.5 hour on a PC 866 Mhz.

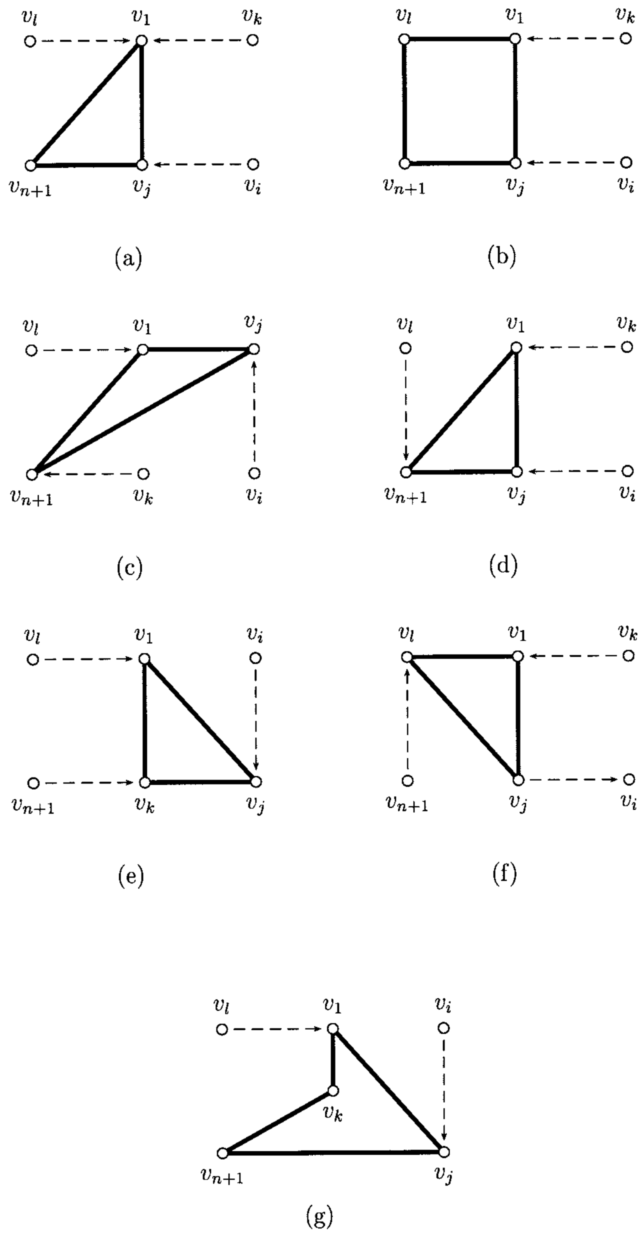

FIG. A.1. RSP solutions for the proof of Proposition 3.5.

Table 4 shows that for random instances in Class III, that is, those generated as in [13, 18], we were able to solve to optimality 122 of 130 instances, involving up to 300 vertices. The largest instances that we were able to solve contain three times as many vertices as those solved by Lee et al. [13], who considered a closely related problem. The better performance of our code is due mainly to our exact sepa-

TABLE A.1. Affinely independence certificate for Proposition 3.5.

Points	$y_{n+1,1}$	$y_{n+1,n+1}$	$y_{n+1,l} : l \neq 1, n+1$	$y_{l,n+1} : l \neq n+1$	$x_e : e \in \delta(n+1)$
P	$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$	0	0	0	
(a), (b)	$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$	$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$	0	0	$\begin{matrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & 0 & 1 & & \vdots \\ \vdots & \vdots & & & \ddots & 0 \\ 1 & 0 & 0 & \cdots & 0 & 1 \end{matrix}$
(g)	0	1	0	0	$0 \quad 1 \quad 1 \quad 0 \quad \cdots \quad 0 \quad 0$
(c), (d)	$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$	$\begin{matrix} 1 \\ \vdots \\ 1 \end{matrix}$	0	$\begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{matrix}$	$\cdots$
(e), (f)	$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}$	$\begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{matrix}$	0	0

ration of constraints (6) rather than to the separation of the weaker constraints (17). The separation of constraints (14) does not contribute in a sensitive way to the resolution of the instances of Class III, as observed in the last two columns of Table 4, which give the average gap of the LP-relaxation and the heuristic approach at the end of the root node when valid inequalities (14) are not activated. We see that the lower bound and the current best solution are also worse, due to the reduction in the number of iterations (i.e., executions of Step 4). Nevertheless, this worsening of the bounds does not significantly affect the overall execution time since deactivating these constraints saves computational effort in the separation phase. In other words, from our experiments, we conclude that constraints (14) are not a fundamental ingredient of our branch-and-cut approach when solving Class III instances.

Overall, we observe that the RSP is more difficult to solve when an optimal ring contains about 20% of the vertices, which is due mainly to the trade-off between ring and assignment costs. Indeed, instances of classes I and II are easier when α is smaller (thus, the objective function is less sensitive to c_{ij} and, therefore, the optimal cycle is close to be Hamiltonian). The difficulty of solving instances with small optimal rings is due to the relative weakness of the location relaxation.

Comparing Class II ($\alpha = 5$) with Class III, one also observes the impact of the "hub-selection fixed cost" d_{ii}. More precisely, the ring and assignment costs are comparable in both families of instances, but $d_{ii} = 0$ in Class II. This explains why smaller optimal cycles were obtained (see the p^* columns) and why larger instances were solved.

6. CONCLUSIONS

We have presented what we believe to be the first exact algorithm for the *RSP*, a problem arising in several network design contexts. An mixed-integer linear programming formulation including several classes of facet-defining inequalities was proposed, together with a branch-and-cut algorithm. The proposed approach was tested on three classes of instances. The largest solved involved 300 vertices. Recall that two of the classes are the benchmark instances in Moreno Pérez et al. [14], while the third set of instances were generated as those used in Lee et al. [13] and Xu et al. [18].

Acknowledgments

The authors thank the associate editor and three referees for their valuable comments.

APPENDIX

Proof of Proposition 3.5. This proof is obtained using induction on the number $n := |V|$ of vertices. By using the software PORTA [3], we were able to enumerate all facet-defining inequalities for $n = 5$ (see Rodríguez Martín [17]). PORTA is a collection of routines to analyze polyhedra; in particular, it computes all the facet-defining inequalities for the convex hull of a set of points. This allows us to state that the result holds true for $n = 5$.

Assume now that (15) is facet-defining for polytope P associated with a mixed graph $G = (V, E, \cup A)$, with $V = \{v_1, \ldots, v_n\}$, $n \geq 5$. Let $v_{n+1} \notin V$ and define P' as the polytope associated to $G' = (V', E' \cup A')$, where $V' = V \cup \{v_{n+1}\}$, $E' = E \cup \{[v_{n+1}, v_i] : i = 1, \ldots, n\}$, and $A' = A \cup \{(v_{n+1}, v_i), (v_i, v_{n+1}) : i = 1, \ldots, n\}$. First, consider $\dim(P)$ affinely independent vectors of P satisfying (15) at equality. These vectors can be transformed into vectors of P' by adding arc (v_{n+1}, v_1) to the corresponding RSP solution graphs. Further, those $\dim(P)$ new

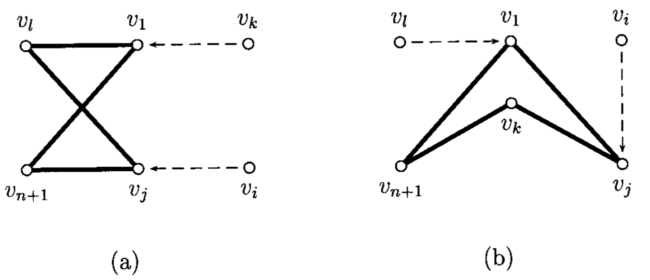

FIG. A.2. RSP solutions for the proof of Proposition 3.6.

vectors are still affinely independent and tight for (15). To complete the proof, we need to exhibit $\dim(P') - \dim(P) = 3n - 2$ affinely independent points of P', tight for (15) and such that $y_{n+1,1} = 0$. Each mixed graph of Figure A.1(a,c,e) provides three such points (obtained by permuting v_i, v_j, and v_k). This gives nine points. Next, each mixed graph of Figure A.1(b,d,f) provides $n - 4$ such points (one for each $l \neq 1, i, j, k$). Finally, Figure A.1(g) gives the last point. To see that all these $3n - 2$ points are affinely independent, it suffices to check that the new components of their characteristic vectors, that is, corresponding to the addition of v_{n+1} to the graph, form a nonsingular matrix (see Table A.1). ∎

Proof of Proposition 3.6. The proof is similar to that of Proposition 3.5.

When $n = 7$, we can verify that (16) is facet-defining by using PORTA [3] (see Rodríguez Martín [17]). Further, the $\dim(P)$ affinely independent vectors of P can be extended exactly in the same way.

When $v_{n+1} \in S$, the $3n - 2$ additional affinely independent vectors illustrated in Figure A.1 are also tight for (16). When $v_{n+1} \notin S$, Figure A.1(b) yields $n - |S| - 1$ vectors tight for (16), that is, only if $v_l \notin S$ and $v_l \neq v_1$, v_{n+1}. Figure A.2(a) provides $|S| - 3$ additional tight vectors, that is, for $v_l \in S \backslash \{v_i, v_j, v_k\}$. Further, Figure A.1(g) must be replaced by Figure A.2(b). ∎

REFERENCES

[1] P. Avella and A. Sassano, On the p-median polytope, Math Program A 89 (2001), 395–411.

[2] P. Bauer, The circuit polytope: Facets, Math Oper Res 22 (1997), 110–145.

[3] T. Christof and A. Loebel, POlyhedron Representation Transformation Algorithm (PORTA), http://www.iwr.uni-heidelberg.de/groups/comopt/software/PORTA.

[4] L. Cooper, Location-allocation problems, Oper Res 11 (1963), 331–343.

[5] J.R. Current and D.A. Schilling, The median tour and maximal covering tour problems: Formulation and heuristics, Eur J Oper Res 73 (1994), 114–126.

[6] M. Fischetti, J.J. Salazar, and P. Toth, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem, Oper Res 45 (1997), 378–394.

[7] M. Gendreau, G. Laporte, and F. Semet, The covering tour problem, Oper Res 45 (1997), 568–576.

[8] E. Gourdin, M. Labbé, and H. Yaman, "Telecommunication and location," Facility location: Applications and theory, Z. Drezner and H.W. Hamacher (Editors), Springer, Berlin, (to appear).

[9] M. Grötschel and M.W. Padberg, , "Polyhedral theory," The traveling salesman problem: A guided tour of combinatorial optimization, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (Editors), Wiley, Chichester, 1995, pp. 251–305.

[10] M. Jünger and S. Thienel, Introduction to ABACUS—A Branch-And-CUt System, Technical report No. 97.263, Universitat zu Köln, 1997.

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

[12] M. Labbé, G. Laporte, and I. Rodríguez Martín, "Path, tree and cycle location," Fleet management and logistics, T.G. Crainic and G. Laporte (Editors), Kluwer, Boston, 1998, pp. 187–204.

[13] Y. Lee, S.Y. Chiu, and J. Sanchez, A branch and cut algorithm for the Steiner ring star problem, Int J Manag Sci 4 (1998), 21–34.

[14] J.A. Moreno Pérez, J.M. Moreno Vega, and I. Rodríguez Martín, Variable neighborhood tabu search and its application to the median cycle problem, Eur J Oper Res (to appear).

[15] M.W. Padberg, On the facial structure of set packing polyhedra, Math Program 5 (1973), 199–215.

[16] G. Reinelt, TSPLIB—A traveling salesman problem library, ORSA J Comput 3 (1991), 376–384.

[17] I. Rodríguez Martín, Cycle location problems, Ph.D. thesis, University of La Laguna, 2000.

[18] J. Xu, S.Y. Chiu, and F. Glover, Optimizing a ring-based private line telecommunication network using tabu search, Manag Sci 45 (1999), 330–345.