THE STACKER CRANE PROBLEM AND THE DIRECTED GENERAL ROUTING PROBLEM

Thais Ávila, Ángel Corberán, Isaac Plana
Dept. d'Estadística i Investigació Operativa, Universitat de València, Spain

José M. Sanchis
Dept. de Matemática Aplicada, Universidad Politécnica de Valencia, Spain

October 23, 2014

Abstract

In this paper we deal with the polyhedral description and the resolution of the Directed General Routing Problem (DGRP) and the Stacker Crane Problem (SCP). The DGRP, in which the service activity occurs both at some of the nodes and at some of the arcs of a directed graph, contains a large number of important arc and node routing problems as special cases, including the SCP. Here, large families of facet defining inequalities are described. Furthermore, a branch-and-cut algorithm for these problems is presented. Extensive computational experiments over different sets of DGRP and SCP instances are included. These results prove that our algorithm is among the best solution procedures proposed for both problems.

Key Words: Directed Rural Postman Problem, Directed General Routing Problem, Stacker Crane Problem, branch-and-cut algorithm.

1 Introduction

The Stacker Crane Problem (SCP) basically consists of finding a tour that starts and ends at a given vertex and traverses a set of arcs of a mixed graph with minimum cost. Its name refers to the practical problem of operating a crane. The crane must start from an initial position, perform a set of movements, and return to the initial position. The objective is to schedule the movements of the crane so as to minimize the total tour cost. This problem can be considered an arc routing problem, particularly a special case of the Directed Rural Postman Problem, a Pickup and Delivery Problem, or a special case of the Asymmetric Traveling Salesman Problem (ATSP).

The SCP was proposed by Frederickson, Hecht, and Kim [11], who distinguished three versions of the stacker crane problem by specifying where the crane terminates. For the first version, the crane must return to the depot. For the second one, the crane stops at a particular vertex, not necessarily the depot, while for the third one the crane is allowed to finish at any vertex.

In this paper we will refer to the first version of the SCP, which can be defined as an arc routing problem on a mixed graph $G = (V, E, A)$, where V is the set of vertices, E the set of edges, and A the set of arcs. Each link (arc or edge) of the graph has an associated non-negative cost. The goal is to find a minimum cost tour, starting and finishing at the depot, which traverses all the arcs in A.

Frederickson, Hecht, and Kim ([11], [12]) showed that the SCP is NP-hard, by proving that any instance of the Traveling Salesman Problem (TSP) can be transformed into an SCP one. They also developed a heuristic algorithm with a worst-case performance ratio of 9/5 and $O(n^3)$ complexity. This procedure needs the graph G to satisfy that each vertex is either the head or the tail of exactly one arc in A, that the cost of an arc between two vertices is not less than the cost of an edge between them, and that edge costs satisfy the triangle inequality.

Berbeglia et al. [1] presented the SCP as a pickup and delivery problem with a unit capacity vehicle. This is an important class of vehicle routing problems in which commodities or people have to be transported from origins to destinations. They have been the object of study in recent years because of their many applications in logistics, ambulatory services, and robotics. Eiselt, Gendreau, and Laporte [10] presented a survey on the RPP and devoted a section to the SCP. Zhang [20] proposed a simplification of the algorithm in [12] with a worst-case ratio of 2 and $O(n^2)$ complexity. Zhang and Zheng [21] and Laporte [15] solved the SCP as an Asymmetric Traveling Salesman Problem (ATSP). Hassin and Khuller [13] proposed a $\frac{1}{2}$ z-approximation algorithm for the ATSP and used it to solve the SCP. Recently, Srour and Velde [19] studied the difficulty of the SCP presenting an statistical study comparing the difficulty of the resolution of SCP instances with that of general ATSP instances.

As it will be seen later, the SCP is a special case of the Directed General Routing Problem (DGRP), a problem in which the service activity occurs both at some of the vertices and some of the arcs of a directed graph. More formally, let $G = (V, A)$ be a directed graph and consider a subset of vertices $V_R \subseteq V$ and a subset of arcs $A_R \subseteq A$. The DGRP consists of finding a minimum cost tour visiting all the vertices in V_R and traversing all the arcs in A_R. Blais and Laporte [2] transformed the DGRP into an equivalent ATSP and solved it by using the exact algorithm proposed by Carpaneto, Dell'Amico, and Toth [4]. With this procedure, Blais and Laporte were able to solve very large DGRP instances to optimality. Although there is no previous work addressing the DGRP directly as an arc routing problem, several papers have been devoted to the study and resolution of the General Routing Problem defined on a mixed graph (MGRP). Particularly, in Corberán, Romero, and Sanchis [9] and Corberán, Mejía, and Sanchis [7], a formulation of the problem and a partial description of its associated polyhedron was presented. Furthermore, the authors proposed a cutting-plane algorithm producing good computational results. Most of the results presented there can be directly applied to the DGRP by considering $E = \emptyset$. Two early works on the Directed Rural Postman Problem (DRPP), which can be considered as a special case of the DGRP when $V_R = \emptyset$ are those by Savall [17] and Campos and Savall [3]. In the first work, a preliminary polyhedral study of the DRPP is proposed, as well as several heuristic algorithms for its resolution that were improved and published later in [3].

In all the papers above, it is assumed, without loss of generality, that the original graph G has been transformed to satisfy that $V = V_R$. This is not a serious restriction as there is a simple way to transform arc routing instances which do not satisfy the assumption into instances which do (see, for instance, Christofides et al. [5] or Eiselt, Gendreau, and Laporte [10]. Such a transformation, which eliminates the non-required vertices, makes easier both the formulation of the problem and the implementation of the algorithms, but sometimes the transformed graph could have many more arcs than the original one, making the problem harder to solve. In this paper, we study the SCP and the DGRP defined on the original graph, in which condition $V = V_R$ does not need to be satisfied. We present a formulation for the DGRP, study its associated polyhedron, and implement a branch-and-cut algorithm that is able to solve large-sized DGRP and SCP instances to optimality.

More precisely, in Section 2 both problems are defined and modeled, and some notation is introduced. Section 3 is devoted to the polyhedral study of the DGRP, where different families of valid and facet-inducing inequalities are described. The branch-and-cut algorithm and the computational results obtained on different sets of SCP and DGRP instances are presented in Section 4.

2 Problems definition and notation

Consider a mixed graph $G = (V, E, A)$ with set of vertices V, set of edges E, and set of arcs A. Let $1 \in V$ be the depot. Associated with each arc and edge (i, j) there is a nonnegative cost c_{ij}. The Stacker Crane Problem consists of finding a closed walk starting and finishing at the depot, traversing each arc in A, and such that the cost of the tour is minimum.

In general, the graph induced by A and the depot is a disconnected graph. We will call p to the number of its connected components, $V_1, \dots, V_p$ will denote the corresponding vertex sets (called R-sets), and $V_R = V_1 \cup ... \cup V_p$ the set of required vertices, that is, the set of vertices that are incident with arcs plus the depot. Note that, if the depot is not incident with any arc, one of these R-sets will consists of just the depot. Hence, we have a strongly connected mixed graph $G = (V, A, E) := (V_R \cup V_{NR}, A, E)$, where $V_{NR} = V \setminus V_R$. For any vertex $i \in V$, $d^+(i)$ and $d^-(i)$ represent the number of arcs leaving and entering i, respectively. Given a subset of vertices $S \subset V$, we define $\delta(S) = \{(i, j) \in E \ : \ i \in S, \ \ j \in V \setminus S\}$. For simplicity, we will use $\delta(i)$ instead of $\delta(\{i\})$.

A closed walk starting and ending at the depot v_1 that traverses exactly once each arc in A is called a *tour for the SCP*. Similarly, a *semitour for the SCP* is the subset of edges obtained after removing the arcs from a given tour for the SCP. Given that the traversal of the arcs is common to all the solutions, we can formulate the SCP in terms of the semitours. For each edge $e = (i, j)$, let x_{ij} and x_{ji} be the number of times edge e is traversed from i to j and from j to i, respectively. The SCP can be formulated as

$$\text{Minimize} \quad \sum_{(i,j)\in E} c_{ij}(x_{ij} + x_{ji}) \tag{1}$$

$$\sum_{(i,j)\in\delta(i)} (x_{ij} - x_{ji}) = d^-(i) - d^+(i), \qquad \forall i \in V \tag{2}$$

$$\sum_{i\in S,\ j\notin S} x_{ij} \geq 1, \qquad \forall S = \big(\bigcup_{k\in Q} V_k\big) \cup W, \quad Q \subsetneq \{1, \dots, p\}, \ W \subseteq V_{NR} \tag{3}$$

$$x_{ij}, x_{ji} \geq 0 \ \text{ and integer}, \qquad \forall (i, j) \in E \tag{4}$$

Constraints (2) are the symmetry conditions on the vertices. Given that the R-sets are connected subgraphs, constraints (3) force the solution to be connected. Vectors $x \in \mathbb{R}^{2|E|}$ satisfying (2) to (4) correspond to the semitours for the SCP, guaranteeing that, after adding the arcs in A, the graph obtained will be symmetric and strongly connected and, hence, an Eulerian graph.

Using two variables for each edge, representing the number of times it is traversed in each direction, is equivalent to transforming each edge into two opposite arcs with the same cost. Hence, the SCP is a special case of the Directed General Routing Problem, DGRP. In what follows, we will study the more general problem.

As mentioned before, given a directed graph $G = (V, A)$, a set of required vertices $V_R \subseteq V$

and a set of required arcs $A_R \subseteq A$, the DGRP consists of finding a minimum cost tour visiting all the required vertices and traversing all the required arcs. In the DGRP, it is helpful to assume, without loss of generality, that the vertices incident to any required arc are also in V_R. Hence, we have a strongly connected graph $G = (V, A) := (V_R \cup V_{NR}, A_R \cup A_{NR})$, where $V_{NR} = V \setminus V_R$ and $A_{NR} = A \setminus A_R$.

The graph induced by V_R and A_R is in general a disconnected graph. As with the SCP, we will call p to the number of its connected components and $V_1, \dots, V_p$ will denote the corresponding vertex sets, where $V_1 \cup ... \cup V_p = V_R$. Note that, some of these R-sets can consist of just one single vertex. For any vertex $i \in V$, $d_R^+(i)$ and $d_R^-(i)$ represent the number of required arcs leaving and entering i, respectively. Given two subsets of vertices $S_1, S_2 \subset V$, we define $A(S_1 : S_2) = \{(i,j) \in A \ : \ \ i \in S_1, j \in S_2\}$ and $(S_1 : S_2) = A(S_1 : S_2) \cup A(S_2 : S_1)$. Given $S \subset V$, $\delta^+(S) = A(S : V \setminus S)$, $\delta^-(S) = A(V \setminus S : S)$, and $A(S) = \{(i,j) \in A \ : \ \ i \in S, j \in S\}$. Subsets $A_R(S_1 : S_2)$, $(S_1 : S_2)_R$, etc, and $A_{NR}(S_1 : S_2)$, $(S_1 : S_2)_{NR}$, etc, refer to the required and non-required arcs, respectively, of the above defined subsets.

If we define variables x_{ij} as the number of times that arc (i,j) is traversed in deadheading by the solution, the DGRP can be formulated, in terms of semitours, as

$$\text{Minimize} \quad \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{5}$$

$$x(\delta^+(i)) - x(\delta^-(i)) = d_R^-(i) - d_R^+(i), \qquad \forall i \in V \tag{6}$$

$$x(\delta^+(S)) \geq 1, \qquad \forall S = \big(\bigcup_{k\in Q} V_k\big) \cup W, \ \ Q \subsetneq \{1,\dots,p\}, \ W \subseteq V_{NR} \tag{7}$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A \tag{8}$$

$$x_{ij} \text{ integer}, \qquad \forall (i,j) \in A \tag{9}$$

where given a subset $F \subseteq A$, $x(F) = \sum_{(i,j)\in A} x_{ij}$. Conditions (6) are the symmetry equations and conditions (7) are the connectivity inequalities. Note that any $|V|-1$ of the equations in (6) are linearly independent.

3 DGRP Polyhedron

Let DGRP(G) be the convex hull of all the semitours $x \in \mathbb{Z}^{|A|}$ satisfying (6) to (9). With a similar proof to that in Corberán, Romero, and Sanchis [9] for the MGRP, it can be seen that DGRP(G) is an unbounded polyhedron of dimension $|A| - |V| + 1$ if G is a strongly connected graph, and that the trivial inequalities $x_{ij} \geq 0$ are facet-inducing $\forall (i,j) \in A$ such that $G \setminus \{(i,j)\}$ is strongly connected.

All other facet-inducing inequalities for the DGRP(G) are *configuration inequalities* (Naddef and Rinaldi [16]). A *configuration* $\mathcal{C}$ on G is a pair $(\mathcal{B}, c)$, where $\mathcal{B} = \{B_1, B_2, \dots, B_r\}$ is a partition of V and c is a real function defined on $\mathcal{B} \times \mathcal{B}$ satisfying that every subgraph $G(B_i)$ is strongly connected and there is no closed cycle $B_p, B_q, \dots, B_m, B_p$ with total c-cost negative.

Associated with a configuration, there is a *configuration graph*, $G_{\mathcal{C}}$. This graph has node set $\mathcal{B}$, of which those with $B_p \cap V_R \neq \emptyset$ are considered required nodes, a required arc (B_i, B_j) for each required arc (u,v) of G with $u \in B_i$, $v \in B_j$, and a non-required arc (B_i, B_j) for each pair B_i, B_j such that $A_{NR}(B_i : B_j) \neq \emptyset$. In other words, $G_{\mathcal{C}}$ is the graph resulting after shrinking node sets B_i, $i = 1, \dots, r$, into a single vertex each, and shrinking each set of non-required parallel arcs into one single arc, but keeping all the required arcs.

A configuration $\mathcal{C}$ defines a *configuration inequality*, $\sum_{(i,j)\in A} c_{ij}x_{ij} \geq c_0$, where:

- $c_{ij} = 0$ for every $(i,j) \in A(B_q)$, $q = 1,\ldots,r$
- $c_{ij} = c(B_p, B_q)$ for every arc $(i,j) \in A(B_p : B_q)$
- c_0 is the c-length of the shortest semitour for the DGRP on $G_{\mathcal{C}}$ (and on G).

Like in Corberán, Romero, and Sanchis [9], it can be proved that all facet-inducing inequalities for DGRP(G), except those equivalent to trivial ones, are configuration inequalities. Note that a configuration $\mathcal{C}$ on G can also be considered a configuration on the shrunk graph $G_{\mathcal{C}} = (\mathcal{B}, \mathcal{A}_{\mathcal{C}})$ and therefore defines also an inequality on this graph:

$$\sum_{(i,j)\in A_{\mathcal{C}}} c(B_i, B_j)x_{ij} \geq c_0,$$

where x_{ij} denotes the number of times arc (B_i, B_j) is traversed.

Note 1 Any semitour x for the DGRP in G can be shrunk into a semitour $x_{\mathcal{C}}$ for the DGRP in $G_{\mathcal{C}}$ with the same c-cost. Moreover, since subgraphs $G(B_i)$ are strongly connected, any semitour $x_{\mathcal{C}}$ for the DGRP in $G_{\mathcal{C}}$ can be extended to a semitour x for the DGRP in G, also with the same c-cost. Therefore, if the configuration inequality is valid for DGRP$(G_{\mathcal{C}})$, it will also be valid for DGRP(G). Furthermore, the following 'lifting' theorem states that a given configuration inequality which is facet-inducing for DGRP$(G_{\mathcal{C}})$ is also facet-inducing for DGRP(G).

Theorem 1 *Let G be a directed graph and let $\mathcal{C}$ be a configuration on G. The associated configuration inequality is facet-inducing for DGRP(G) if the configuration inequality associated with $\mathcal{C}$ on graph $G_{\mathcal{C}}$ is facet-inducing for DGRP($G_{\mathcal{C}}$).*

Proof: The proof is similar to the one in [9] and is omitted here for the sake of brevity. ♦

In what follows, we will prove that the connectivity inequalities and other families are facet-inducing for DGRP(G).

Theorem 2 : *Inequalities (7), $x(\delta^+(S)) \geq 1, \forall S = \big(\bigcup_{k\in Q} V_k\big) \cup W$, $Q \subsetneq \{1,\ldots,p\}$, $W \subseteq V_{NR}$, are facet-inducing for DGRP(G) if graphs $G(S)$ and $G(V \setminus S)$ are strongly connected.*

Proof: The configuration graph $G_{\mathcal{C}}$ has only two nodes, say B_1 and B_2, corresponding to S and to $V \setminus S$, respectively (both of them required) and a pair of opposite non-required arcs. Therefore, dim(DGRP$(G_{\mathcal{C}})$)$= 2 - 2 + 1 = 1$ and, since the semitour $x_{B_1B_2} = x_{B_2B_1} = 1$ satisfies $x(\delta^+(S)) = x_{B_1B_2} = 1$, the inequality is facet-inducing for DGRP$(G_{\mathcal{C}})$ and, therefore, is also facet-inducing for DGRP(G). ♦

K-C inequalities

A *K-C configuration* (see Figure 1) is defined by an integer $K \geq 3$, a partition of V into $K+1$ subsets $\{M_0, M_1, \ldots, M_K\}$ such that each R-set V_i, $1 \leq i \leq p$, is contained in exactly one of the node sets $M_0 \cup M_K, M_1, \ldots, M_{K-1}$, each node set $M_0 \cup M_K, M_1, \ldots, M_{K-1}$ contains at least one

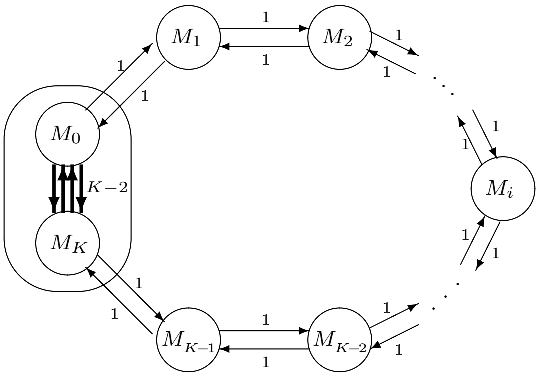

Figure 1: K-C configuration.

R-set, the induced subgraphs $G(M_i)$, $i = 0, 1, \ldots, K$, are strongly connected, and $(M_0 : M_K)$ contains a positive number of required arcs, such that $|A_R(M_0 : M_K)| = |A_R(M_K : M_0)|$ and by the cost functions defined as $c(M_0, M_K) = c(M_K, M_0) = K - 2$ and $c(M_i, M_j) = |i - j|$, $\forall i, j \ : \ \{i, j\} \neq \{0, K\}$. The partition $\mathcal{B} = \{M_0, M_1, \ldots, M_K\}$ and the above costs define the configuration graph $G_{\mathcal{C}}$ whose skeleton is shown in Figure 1. Arcs (M_i, M_j) not represented in Figure 1 have a cost equal to the length of the shortest path from M_i to M_j using arcs in the skeleton. The associated *K-C inequality* is

$$(K-2)\; x\Big((M_0 : M_K)\Big) + \sum_{\substack{0 \le i < j \le K \\ (i,j) \neq (0,K)}} |\, i-j \,|\; x\Big((M_i : M_j)\Big) \;\geq\; 2(K-1). \tag{10}$$

Theorem 3 *K-C inequalities* (10) *are valid and facet-inducing for DGRP(G).*

Proof: Since all the nodes in the K-C configuration graph $G_{\mathcal{C}}$ are required, it is a special case of the K-C configuration graph for the MGRP presented in [9]. Hence, the corresponding K-C inequality is valid and facet-inducing for DGRP($G_{\mathcal{C}}$) and, from Note 1 and Theorem 1, it follows that it is also valid and facet-inducing for DGRP(G). ♦

Honeycomb inequalities

Honeycomb inequalities are a generalization of K-C inequalities. In a K-C configuration (see figure 1), a R-connected component (or a cluster of R-connected components) is divided into two parts (M_0 and M_K). In this section we generalize this configuration simultaneously both in the number of parts a R-connected component is divided into and in the number of R-connected components we divide.

Consider a partition of the set of vertices V into K vertex sets $\{M_1, \ldots, M_L, M_{L+1}, \ldots, M_K\}$, $3 \leq K \leq p$, $1 \leq L \leq K$, in such a way that each R-set V_j is contained in exactly one M_i, each node set M_i contains at least one R-set, and the induced subgraphs $G(M_i)$ are strongly connected. Suppose we can now partition each set M_i, $i = 1, \ldots, L$, into $\gamma_i \geq 2$ subsets, $M_i = B_i^1 \cup \ldots \cup B_i^{\gamma_i}$, satisfying the following conditions:

H1) Each B_i^j contains an even number of R-odd nodes, $j = 1, 2, \ldots, \gamma_i$.

H2) The induced subgraphs $G(B_i^j), j = 1, 2, \ldots, \gamma_i$, are strongly connected.

H3) The graph with node set $B_i^1, \dots, B_i^{\gamma_i}$ and having an arc (B_i^j, B_i^k) for each required arc $a \in A_R(B_i^j : B_i^k)$, is symmetric and connected.

For notational convenience, we denote $B_i^0 = M_i$, $i = L+1, \dots, K$. We have therefore the following partition of V:

$$\mathcal{B} = \{B_1^1, \dots, B_1^{\gamma_1},\ \dots\ , B_L^1, \dots, B_L^{\gamma_L},\ B_{L+1}^0, \dots, B_K^0\}$$

This partition $\mathcal{B}$ defines a configuration graph $G_{\mathcal{C}} = (\mathcal{B}, \mathcal{A})$ with a set of nodes $\mathcal{B}$ and a set of arcs $\mathcal{A}$ formed by a required arc (B_r^i, B_q^j) for each required arc $a \in A_R(B_r^i : B_q^j)$ and a non-required arc (B_r^i, B_q^j) between each couple of nodes B_r^i, B_q^j such that $A_{NR}(B_r^i : B_q^j) \neq \emptyset$.

Let us suppose that there is a set T of pairs of opposite non-required arcs in $G_{\mathcal{C}}$ joining nodes corresponding to different M_j, $j = 1, \dots K$, such that the undirected graph with node set $\mathcal{B}$ and having an edge (B_i^j, B_p^q) for each pair of opposite arcs (B_i^j, B_p^q), (B_p^q, B_i^j) in T, is a spanning tree. Then, for each pair of nodes B_i^j, B_p^q in $\mathcal{B}$, $d(B_i^j, B_p^q)$ will denote the number of arcs in the unique path in $(\mathcal{B}, T)$ from B_i^j to B_p^q. We will assume that the following condition is also satisfied:

H4) $d(B_i^j, B_i^q) \geq 3 \quad \forall\ i = 1, \dots, L$ and $\forall\ j \neq q$.

The graph $(\mathcal{B}, T)$ defines the skeleton of the configuration (see Figure 2, where the arcs in T are represented in thin lines and the required arcs in bold lines). We assume that

H5) the undirected graph $(\bar{\mathcal{M}}, T_{\bar{\mathcal{M}}})$, with node set $\bar{\mathcal{M}} = \{M_1, \dots, M_L, M_{L+1}, \dots, M_K\}$ and having an edge (M_i, M_j) for each pair of opposite arcs (B_i^p, B_j^q), (B_j^q, B_i^p) in T is 2-connected, and

H6) the indegree and outdegree of every node B_q^i, $i \neq 0$, in $(\mathcal{B}, T)$ is equal to 1.

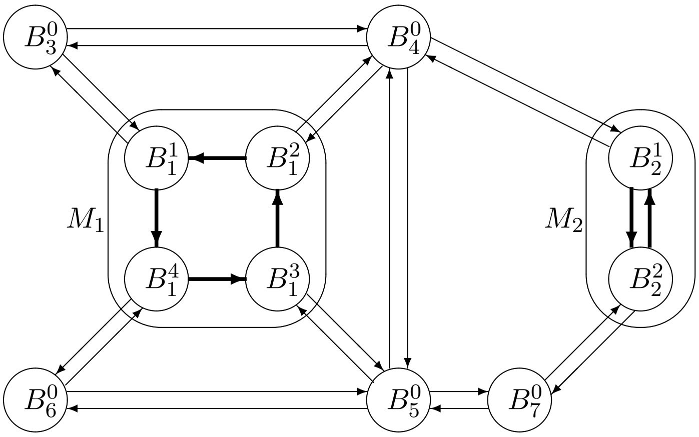

Figure 2: Honeycomb Configuration

We define the configuration costs on the arcs of $(\mathcal{B}, \mathcal{A})$ as follows.

- For the arcs (B_q^i, B_q^j): $c(B_q^i, B_q^j) = d(B_q^i, B_q^j) - 2$.
- For the arcs (B_r^i, B_q^j), $r \neq q$: $c(B_q^i, B_q^j) = d(B_q^i, B_q^j)$.

The *Honeycomb inequality* corresponding to this configuration is defined by

$$\sum_{(i,j) \in A} c_{ij} x_{ij} \geq 2(K-1). \tag{11}$$

Theorem 4 *Honeycomb inequalities* (11) *are valid and facet-inducing for DGRP(G).*

Proof: Since all the nodes in the Honeycomb configuration graph $G_\mathcal{C}$ are required, it is a special case of the Honeycomb configuration graph for the MGRP presented in [7]. Hence, the corresponding Honeycomb inequality is valid and facet-inducing for DGRP($G_\mathcal{C}$) and, from Note 1 and Theorem 1, it follows that it is also valid and facet-inducing for DGRP(G). ♦

Path-Bridge inequalities

Like Honeycomb inequalities, Path-Bridge inequalities are also a generalization of K-C inequalities. However, the generalization is in a different direction and neither class contains the other.

A *Path-Bridge configuration* (Figure 3a) is defined by two integers P (the number of paths) and B (the arcs in the bridge) with $P \geq 1$, $B \geq 0$, $P + B \geq 3$ and odd, by $n_i \geq 2$ integers, $i = 1, \ldots, P$, and a partition of V into subsets $\{M_0, M_Z, M_j^i : i = 1, \ldots P,\ j = 1, \ldots n_i\}$. The partition must satisfy that

- each R-set V_i is contained in exactly one of the node sets $M_0 \cup M_Z$, M_j^i, $i = 1, \ldots P$, $j = 1, \ldots n_i$ (i.e., each required arc either lies in some $A(M_j^i)$ or crosses from M_0 to M_Z),
- each node set $M_0 \cup M_Z$, M_j^i, $i = 1, \ldots P$, $j = 1, \ldots n_i$ contains at least one R-set,
- the induced subgraphs $G(M_j^i)$, $i = 0, 1, \ldots, P$, $j = 1, \ldots n_i + 1$, are strongly connected (where, for convenience, for all i we identify M_0^i with M_0 and $M_{n_i+1}^i$ with M_Z),
- $(M_0 : M_Z)$ contains a number B of required arcs and $|A_R(M_0 : M_Z)| = |A_R(M_Z : M_0)|$. Note that this implies that B is even and P is odd (this condition is not necessary for the MGRP), and
- sets $A(M_j^i : M_{j+1}^i)$ and $A(M_{j+1}^i : M_j^i)$, $i = 1, \ldots, P$, $j = 0, 1, \ldots, n_i$, are nonempty.

The associated costs are defined as

$$\begin{aligned}
&c(M_0, M_Z) = c(M_Z, M_0) = 1 \\
&c(M_j^i, M_q^i) = \frac{|j-q|}{n_i - 1}, \qquad \forall j, q \in \{0, 1, \ldots, n_i + 1\},\quad 0 < |j - q| < n_i + 1 \\
&c(M_j^i, M_q^r) = \frac{1}{n_i - 1} + \frac{1}{n_r - 1} + \left|\frac{j-1}{n_i - 1} - \frac{q-1}{n_r - 1}\right|, \\
&\qquad \forall i, r \in \{1, \ldots, P\},\ i \neq r,\quad j \in \{1, \ldots, n_i\},\ q \in \{1, \ldots, n_r\}
\end{aligned}$$

The partition $\{M_0, M_Z, M_j^i : i = 1, \ldots P,\ j = 1, \ldots n_i\}$ and the costs c define the configuration graph $G_\mathcal{C}$ whose skeleton is showed in Figure 3a. It has P paths from M_0 to M_Z, each of them with $n_i + 2$ nodes and $n_i + 1$ pairs of opposite arcs. Arcs (M_j^i, M_q^i), not represented in Figure 3a, have a cost equal to the length of the shortest path from M_j^i to M_q^i using arcs in the skeleton.

The *Path-Bridge inequality* corresponding to this configuration is defined by

$$\sum_{(i,j) \in A} c_{ij} x_{ij} \geq 1 + \sum_{i=1}^{P} \frac{n_i + 1}{n_i - 1}. \tag{12}$$

Note that when $P = 1$, the PB configuration becomes a K-C configuration.

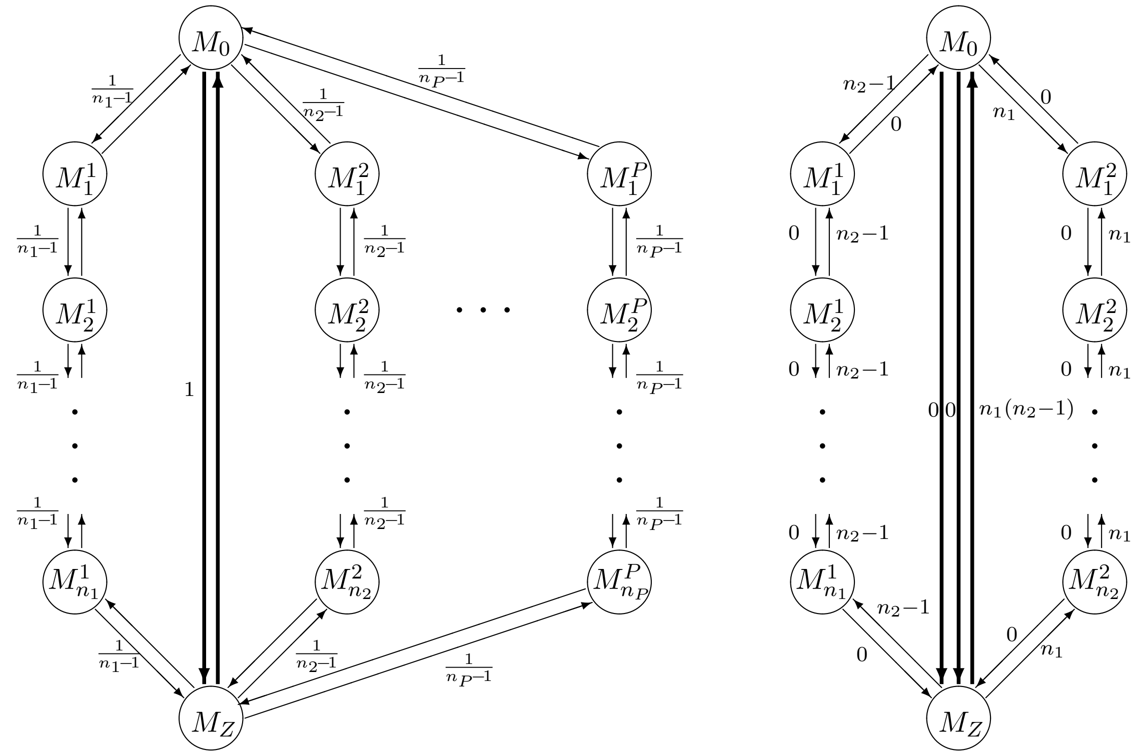

Figure 3: Path-Bridge and Asymmetric 2-Path-Bridge configurations.

Theorem 5 *Path-Bridge inequalities* (12) *are valid and facet-inducing for DGRP(G).*

Proof: The proof is similar to the one in Theorem 3 and is omitted here for the sake of brevity. ⧫

There are other families of inequalities for the MGRP, the K-C$_{02}$, PB$_{02}$ and Honeycomb$_{02}$ inequalities, which are also facet-inducing. It can be seen that these families of inequalities are not facet-inducing for DGRP(G) because they are dominated by the above described (standard) K-C, PB, and Honeycomb inequalities.

New Asymmetric 2-Path-Bridge inequalities

As said above, Path-Bridge inequalities for the DGRP need that condition $|A_R(M_0 : M_Z)| = |A_R(M_Z : M_0)|$ holds and, therefore, P (the number of paths between M_A and M_Z) has to be odd. In this section we present a related family of valid inequalities for the DGRP in which $P = 2$.

An *Asymmetric 2-Path-Bridge configuration* (see Figure 3b) is defined by an odd integer B, $n_1, n_2 \geq 2$ integers, and a partition of V into subsets $\{M_0, M_Z, M_1^1, \dots, M_{n_1}^1, M_1^2, \dots, M_{n_2}^2\}$. As for the (standard) Path-Bridge inequalities, the partition must satisfy that

- each R-set V_i is contained in exactly one of the node sets $M_0 \cup M_Z$, M_j^i,
- each node set $M_0 \cup M_Z$, M_j^i contains at least one R-set,
- the induced subgraphs $G(M_j^i)$ are strongly connected (again, we identify M_0^1 and M_0^2 with M_0 and $M_{n_1+1}^1$ and $M_{n_2+1}^2$ with M_Z),

- sets $A(M_j^i : M_{j+1}^i)$ and $A(M_{j+1}^i : M_j^i)$ are nonempty, and
- $(M_0 : M_Z)$ contains B required arcs satisfying $|A_R(M_0 : M_Z)| = |A_R(M_Z : M_0)| + 1$.

The associated costs are defined as follows (see Figure 3b). For the arcs in the bridge, $c(M_0, M_Z) = 0$ and $c(M_Z, M_0) = n_1(n_2 - 1)$. For the arcs in path 1, $c(M_0, M_1^1) = n_2 - 1$ and $c(M_1^1, M_0) = 0$, while $c(M_j^1, M_{j+1}^1) = 0$ and $c(M_{j+1}^1, M_j^1) = n_2 - 1$ for all $j \geq 1$. For the arcs in path 2, $c(M_0, M_1^2) = n_1$ and $c(M_1^2, M_0) = 0$, while $c(M_j^2, M_{j+1}^2) = 0$ and $c(M_{j+1}^2, M_j^2) = n_1$ for all $j \geq 1$. The costs for the arcs joining nodes of the same path is that of the shortest path between them using arcs in the skeleton. Finally, the costs for the arcs joining nodes on different paths are obtained by sequential lifting, i.e., these arcs are ordered in an arbitrary way $a_1, \ldots, a_h$ and, for $i = 1$ to h, c_{a_i} is the maximum value such that a_i belongs to a semitour of cost $n_1(n_2 - 1) + n_1 n_2$ using only arcs from the skeleton and $\{a_1, \ldots, a_i\}$.

The *Asymmetric 2-Path-Bridge (A2PB) inequality* corresponding to this configuration is defined by

$$\sum_{(i,j)\in A} c_{ij} x_{ij} \geq n_1(n_2 - 1) + n_1 n_2. \tag{13}$$

Note that in the special case in which $n_2 = 1$, the above inequality is $n_1 x\left(\delta^+(M_1^2)\right) \geq n_1$, which is a connectivity inequality (7).

Theorem 6 *Asymmetric 2-Path-Bridge inequalities* (13) *are valid for DGRP(G).*

Proof: It suffices to prove validity in the configuration graph $G_{\mathcal{C}}$. Let $F(x) \geq n_1(n_2-1)+n_1n_2$ be an A2PB inequality. From the definition of the coefficients of the arcs not in the skeleton (obtained either as shortest-path lengths or by sequential lifting), it suffices to prove validity for each semitour x for the DGRP on $G_{\mathcal{C}}$ using only arcs in the skeleton.

Let y be a tour for the DGRP of minimum length in the configuration graph $G_{\mathcal{C}}$. Note that, since all the nodes in the configuration graph are required, y must visit all of them. If y uses exactly once each required arc in $(M_0 : M_Z)$ it is easy to see that it has to be similar to one of the tours depicted in Figure 4, whose corresponding semitours x satisfy $F(x) = n_1(n_2 - 1) + n_1 n_2$. If y traverses in deadheading an arc $a \in A(M_Z : M_0)$ then by replacing a by the arcs in path 1 from M_Z to M_0 we obtain a tour with the same cost. If y traverses in deadheading an arc $a \in A(M_0 : M_Z)$ then, given that $|A_R(M_0 : M_Z)| = |A_R(M_Z : M_0)| + 1$, the traversal of $(M_0 : M_Z)$ is unbalanced by two units and both paths 1 and 2 have to be traversed from M_Z to M_0, with a cost of $n_1(n_2 - 1) + n_1 n_2$. ♦

Theorem 7 *Asymmetric 2-Path-Bridge inequalities* (13) *are facet-inducing of DGRP(G).*

Proof: Again, it suffices to prove the result for the configuration graph $G_{\mathcal{C}}$ (see Theorem 1). Let A' denote the set of arcs that are not in paths 1 or 2. The dimension of DGRP($G_{\mathcal{C}}$) is $|A'| + 2(n_1 + 1) + 2(n_2 + 1) - (n_1 + n_2 + 2) + 1 = |A'| + n_1 + n_2 + 3$ and this is the number of affinely (or, in this case, linearly) independent semitours satisfying inequality (13) as an equality we have to find.

It can be seen that, for any arc a in A', there is a semitour satisfying (13) as an equality and using arc a once and arcs in the paths 1 and 2. The arcs in path 1, respectively 2, are denoted

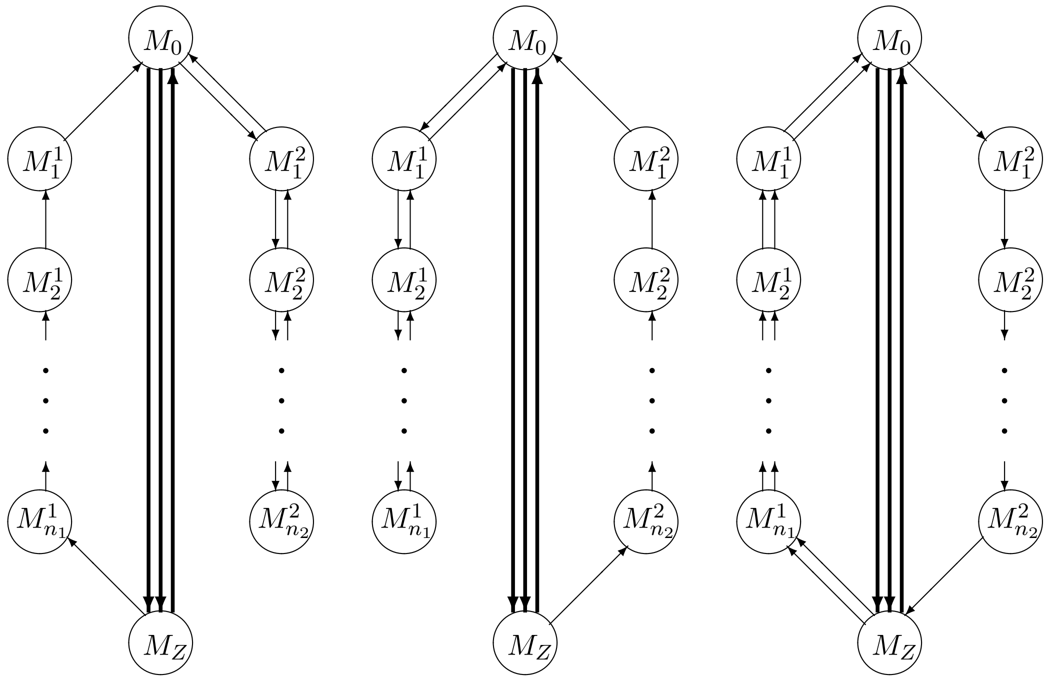

Figure 4: Tours satisfying the A2PB inequality as an equality.

by P1 and P2. In addition, there are $n_2 + 1$ semitours similar to the one depicted in Figure 4(a), $n_1 + 1$ semitours similar to the one depicted in Figure 4(b) and the semitour in Figure 4(c). By expressing these semitours as rows and the arcs as columns, we obtain the matrix in Figure 5(a), where block E and matrix $D_{m\times m}$ are also shown in Figure 5. Note that matrix E contains $n_1 + 1 + n_2 + 1 + 1 = n_1 + n_2 + 3$ rows. It can be proved that matrix E is full rank, which proves the result. ⧫

$$\begin{array}{|c|c|} A' & \text{P1 P2} \\ \hline \text{I} & * \\ \hline 0 & \text{E} \\ \hline \end{array} \qquad E = \begin{array}{cccc} \multicolumn{2}{c}{\text{P1}} & \multicolumn{2}{c}{\text{P2}} \\ \hline D_{n_1+1} & D_{n_1+1} & 0 & 1 \\ 0 & 1 & D_{n_2+1} & D_{n_2+1} \\ 0\dots0 & 2\dots2 & 1\dots1 & 0\dots0 \end{array} \qquad D_m = \begin{pmatrix} 0 & 1 & & 1 & 1 \\ 1 & 0 & & 1 & 1 \\ & & \ddots & & \\ 1 & 1 & & 0 & 1 \\ 1 & 1 & & 1 & 0 \end{pmatrix}_{m\times m}$$

Figure 5: Matrices appearing in the proof of Theorem 7

Although the A2PB inequality using the coefficients obtained by means of sequential lifting is facet-inducing for DGRP(G), these coefficients, except for the first arc, are difficult to find in practice. Therefore, in the separation algorithm we compute the first coefficient and replace the subsequent ones with the costs of the shortest paths using the arcs in the skeleton and this first arc. For the first arc (i, j), if $i \in M_r^1$ and $j \in M_k^2$, its coefficient is given by $c_{ij} = \max\{n_1, (k-1)(n_2-1)+n_1(2-r)\}$. If $i \in M_r^2$ and $j \in M_k^1$, then $c_{ij} = \max\{n_2-1, (n_2-1)(1-k)+n_1(r-1)\}$.

While the inequality obtained with these new coefficients is weaker and may not be facet-inducing, it is a valid inequality and easier to obtain.

4 Computational Experience

We present here the computational results obtained on several sets of DGRP and SCP instances with a branch-and-cut procedure we have implemented based on the polyhedral study presented before. The algorithm has been coded in C/C++ using the Cplex 12.4 MIP Solver with Concert Technology 2.9 on a single thread of an Intel Core i7 at 3.4GHz with 16GB RAM. Cplex heuristic algorithms were turned off, while Cplex own cuts are activated in automatic mode. The optimality gap tolerance was set to zero, and strong branching and the best bound strategies were selected. Finally, Cplex presolve phase is reapplied at the end of the root node, allowing for new iterations of the cutting-plane procedure before branching. All the tests were run with a time limit of one hour.

4.1 The Overall Algorithm

In this section we present a branch-and-cut algorithm that incorporates separation algorithms for the inequalities described in this paper.

4.1.1 Separation algorithms

In this section we present the separation algorithms that have been used to identify the following types of inequalities that are violated by the current LP solution at any iteration of the cutting plane algorithm: connectivity inequalities, K-C inequalities, PB inequalities with 2 paths (2PB), and A2PB inequalities.

Heuristic and exact separation procedures for connectivity inequalities (7) have been adapted from those described in [8] in order to work with graphs containing non-required vertices.

In order to separate K-C, 2PB, and A2PB inequalities, we first apply a shrinking procedure to reduce the size of the support graph, which basically consists of shrinking all the non-required arcs (i, j) for which $x_{ij} = 1$ or $x_{ij} \geq 2$. Then, the heuristic procedures for separating K-C and 2PB inequalities presented in [7] for the Mixed GRP, adapted for the DGRP defined on graphs containing non required vertices, are applied. While applying the separation procedure for 2PB inequalities, we also check for possible A2PB violated inequalities whenever we find an appropriate structure. As previously mentioned, the coefficients of all the arcs joining nodes in different paths, except for the first one, are given by the length of the shortest path using that first arc and those in the skeleton. We choose as the first arc the one with the largest value of x in the solution.

4.1.2 Initial relaxation and cutting-plane algorithm

The initial LP relaxation contains symmetry equations (6), trivial inequalities (8), and a connectivity inequality (7) associated with each R-set.

At each iteration of the cutting plane algorithm the separation procedures are used in the following specific order and the violated inequalities found are added to the LP relaxation:

1. Heuristic separation algorithms for connectivity inequalities.
2. Exact connectivity separation if the corresponding heuristics have failed to find at least 20 violated inequalities.

3. Only at the root node, if no violated connectivity inequalities have been found, heuristic algorithms for separating K-C, 2PB, and A2PB inequalities.

The cutting-plane procedure is applied at each node of the tree until no new violated inequalities are found or a stopping criterium, called *tailing-off*, is satisfied. In our implementation, at the root node the cutting plane stops when the increase in the objective function during the last 20 iterations is less than 0.0001%. At any other node, the cutting plane stops if the increase is less than 0.005% in the last three iterations or if the gap between the lower bound at that node and the global lower bound is greater than 3%.

4.2 Instances and computational results

In this section we present the DGRP and SCP instances that we have used to test the performance of the proposed branch-and-cut algorithm, as well as the computational results obtained.

4.2.1 DGRP instances

We have generated a set of 36 large DGRP instances trying to imitate real street networks. To do that, a number $|V|$ of vertices are randomly generated as points in the 1000×1000 square. For each vertex v, the d shortest edges incident with v are selected and a random direction is assigned to each edge. Rounded Euclidean distances are taken as the arc costs. Then, an arc is declared as required with probability p. All the vertices non incident with required arcs are considered isolated required vertices. Therefore, all these instances satisfy $V = V_R$. An instance is generated for each set of parameters $|V| \in \{500, 750, 1000\}$, $d \in \{3, 4, 5, 6\}$ and $p \in \{0.25, 0.5, 0.75\}$. The name DG742, for example, means that it is a Directed GRP instance with 750 vertices, $d = 4$ and $p = 0.25$. The characteristics and the computational results obtained for the instances with $|V| = 500, 750$, and 1000 are shown in tables 1, 2 and 3, respectively.

Name	$\|V\|$	$\|A_R\|$	$\|A_{NR}\|$	$R-$sets	opt?	Gap0 (%)	Nodes	Time
DG532	500	218	953	286	yes	0.11	3	0.58
DG535	500	443	723	117	yes	0.00	1	0.20
DG537	500	639	529	31	yes	0.00	1	0.14
DG542	500	282	1011	228	yes	0.83	137	22.95
DG545	500	567	712	53	yes	0.00	0	0.11
DG547	500	872	412	6	yes	0.00	0	0.05
DG552	500	317	1072	208	yes	0.00	1	0.23
DG555	500	687	720	21	yes	0.00	0	0.09
DG557	500	959	409	4	yes	0.00	0	0.09
DG562	500	384	1160	146	yes	0.00	0	0.11
DG565	500	804	764	9	yes	0.00	0	0.08
DG567	500	1132	403	1	yes	0.00	0	0.11

Table 1: DGRP instances with $|V|$=500 and $1166 \leq |A| \leq 1535$.

In these tables, the column labeled 'Gap0(%)' shows the gap obtained at the root node of the branch-and-cut tree, computed as $\frac{UB-LB0}{UB} \times 100$, where $LB0$ represents the lower bound at the end of the root node and UB is the cost of the best solution found. The last two columns give the number of nodes of the branch-and-cut tree and the computing time in seconds. As it can be seen, the results obtained in this set of 36 large instances are very good. The gaps obtained at the end of the root node are very tight and all the instances have been solved to optimality in short computing times.

Name	$\lvert V\rvert$	$\lvert A_R\rvert$	$\lvert A_{NR}\rvert$	$R-$sets	opt?	Gap0 (%)	Nodes	Time
DG732	750	339	1451	422	yes	0.70	322	36.69
DG735	750	654	1150	179	yes	0.00	2	1.02
DG737	750	1003	785	35	yes	0.00	0	0.27
DG742	750	407	1497	366	yes	0.15	7	1.73
DG745	750	852	1090	82	yes	0.00	1	0.38
DG747	750	1244	642	5	yes	0.00	0	0.11
DG752	750	516	1570	292	yes	0.01	2	0.69
DG755	750	973	1114	51	yes	0.00	0	0.2
DG757	750	1537	557	2	yes	0.00	0	0.16
DG762	750	584	1784	218	yes	0.00	0	0.27
DG765	750	1204	1172	11	yes	0.00	0	0.17
DG767	750	1673	603	4	yes	0.00	0	0.19

Table 2: DGRP instances with $|V|$=750 and 1788 $\leq |A| \leq$ 2376.

As said in the Introduction, Blais and Laporte [2] solve the DGRP by transforming it into an equivalent ATSP and then solving it by means of the Carpaneto, Dell'Amico, and Toth [4] exact algorithm. In that way, Blais and Laporte solve very large DGRP instances to optimality (see Table 4).

In order to test the performance of our algorithm on instances of similar sizes to those solved by Blais and Laporte, we have generated a set of instances with the same characteristics. As in [2], we have randomly generated graphs with 5000 vertices and 50000 arcs. To ensure the feasibility of each instance, we have generated an undirected Hamiltonian cycle over all vertices and have included the corresponding arcs (in both directions) in the graph. Then, for each graph, a given number of required vertices and arcs have been randomly selected. The arc costs have been randomly generated according to a discrete uniform distribution on $[10, 110]$. The characteristics of these instances can be seen in Table 4, where $|V|$ and $|A|$ are the number of vertices and arcs of the graph, $|V_R|$ and $|A_R|$ give the number of required vertices and arcs,

Name	$\lvert V\rvert$	$\lvert A_R\rvert$	$\lvert A_{NR}\rvert$	$R-$sets	opt?	Gap0 (%)	Nodes	Time
DG132	1000	464	1930	553	yes	0.89	1890	444.23
DG135	1000	870	1472	246	yes	0.05	3	1.88
DG137	1000	1315	1068	43	yes	0.00	0	0.34
DG142	1000	562	2023	466	yes	0.38	25	6.98
DG145	1000	1133	1469	116	yes	0.00	1	1.13
DG147	1000	1643	908	8	yes	0.00	0	0.22
DG152	1000	657	2197	386	yes	0.02	2	0.72
DG155	1000	1283	1501	52	yes	0.00	0	0.36
DG157	1000	2004	820	4	yes	0.00	0	0.25
DG162	1000	753	2373	306	yes	0.00	0	0.45
DG165	1000	1550	1603	26	yes	0.00	0	0.27
DG167	1000	2349	828	4	yes	0.00	0	0.30

Table 3: DGRP instances with $|V|$=1000 and 2342 $\leq |A| \leq$ 3177.

				Blais & Laporte[1]			Branch and Cut	
$\|V\|$	$\|A\|$	$\|V_R\|$	$\|A_R\|$	$\|V_{ATSP}\|$	# opt.	Time	# opt.	Time
5000	50000	1000	1000	2000	5/5	125.6	5/5	31.3
5000	50000	1000	1500	2500	5/5	193.6	5/5	51.8
5000	50000	1000	2000	3000	5/5	280.3	5/5	28.3
5000	50000	1000	2500	3500	4/5	374.9	5/5	21.9
5000	50000	1000	3000	4000	0/5	-	5/5	25.5
5000	50000	1500	1000	2500	5/5	183.1	5/5	37.3
5000	50000	2000	1000	3000	5/5	244.7	5/5	36.7
5000	50000	2500	1000	3500	5/5	314.5	5/5	57.4
5000	50000	3000	1000	4000	4/5	396.8	5/5	50.7
5000	50000	0	3000	3000	5/5	303.0	5/5	12.5
5000	50000	500	2500	3000	5/5	300.0	5/5	19.3
5000	50000	1500	1500	3000	5/5	269.2	5/5	32.5
5000	50000	2500	500	3000	5/5	226.1	5/5	57.8
5000	50000	3000	0	3000	4/5	273.9	5/5	347.2

Table 4: Characteristics of the Blais and Laporte instances and results obtained ([1] Sun Ultra Sparc Station 10 machine).

respectively, and $|V_{ATSP}|$ represents the number of vertices of the transformed graph used by Blais and Laporte. Five instances of each type have been generated.

Originally, we started studying the DGRP and its resolution transforming the graph so that all the vertices in it were required, as it is common practice in most other arc routing problems. This implied computing all the shortest paths between the required vertices and adding arcs representing them. Although, some of these new arcs were redundant and could be removed, the size of the resulting graphs was so huge, that the procedure was unable to solve even the initial LP of most of them. Note that the number of variables associated with non-required arcs would be around nine millions for an instance with $|V_R| + |A_R| = 3000$. This is what motivated us to study and solve the DGRP working on the original graph, containing required and non-required vertices.

The results obtained with our modified algorithm on these instances are shown in Table 4, where they are compared with the results obtained by the transformation procedure by Blais and Laporte. Although the instances used here are not exactly the same ones as those used in [2], the characteristics are the same, and we think the results are comparable. Column *#opt.* shows the number of instances solved to optimality out of five, and column *Time* reports the average computing time. The time reported for the Blais and Laporte procedure includes the time used to transform the original instance into an ATSP one and the resolution time, which was limited to five minutes. Note that the results by Blais and Laporte were obtained on a Sun Ultra Sparc Station 10, which is a considerably slow machine by today's standards. It can be seen that our algorithm has been able to solve all the 70 instances to optimality in short computing times.

4.2.2 SCP instances

Although there have been previous works on the SCP, most of them deal with the problem as a node routing problem and solve it by transforming the SCP into an ATSP. Therefore, the data

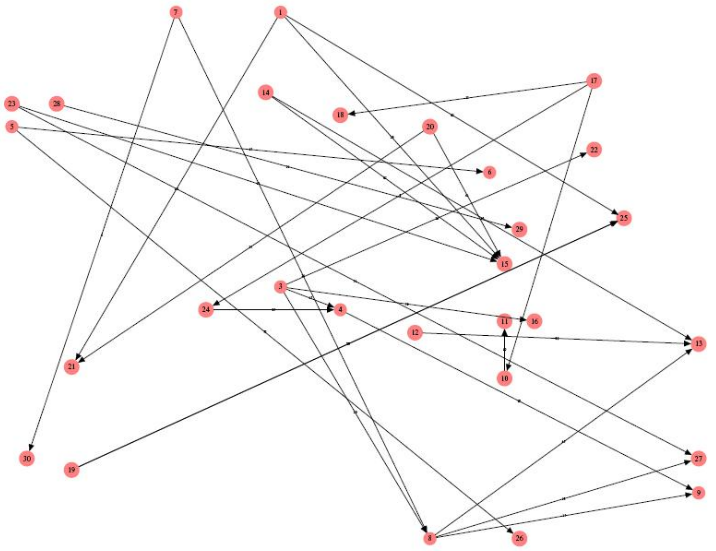

Figure 6: Drayage instance.

of most of the instances available are in ATSP format, i.e. as a matrix giving the distances between the vertices representing the jobs (in this context, required arcs are often referred to as jobs). For this reason we have not been able to test our algorithm on these instances and we have generated two new sets of SCP instances in arc routing format. These new data instances are available at `http://www.uv.es/corberan/instancias`.

The first set of instances have been generated following the procedure described by Srour and van de Velde [19]. These instances are called "drayage" instances because they try to mimic real-life problems arising in the context of drayage transport. According to Srour and van de Velde, these problems are characterized by the fact that "nearly all jobs originate from or are destined to one of only a few fixed freight terminals" and "have an interesting geometric structure" (see Figure 6). The drayage instances generator uses the random number/location generators from the crane generator of Cirasella et al. [6] and Johnson et al. [14]. It selects a set of n points from a $x \times x$ square that will serve as the origins and destinations of n jobs. The first k points are the origins and the last m points the destinations, where k and m are specified by the user. Origins and destinations are then matched in a round-robin fashion until all n jobs have been created. The costs of the arcs are given by Euclidean distances. The code for this generator is available in [18].

Following the above procedure, we have generated 14 drayage instances with the same characteristics as those described in Table C1 in the paper by Srour and van de Velde [19]. These characteristics can be seen in Table 5, which also shows the results obtained with our branch-and-cut algorithm. As it can be seen, all the instances have been solved to optimality in very short computing times. Note also that the gap obtained at the root node of the branch-and-cut tree is very small, even zero for many instances, meaning that the problem has been solved by the cutting-plane procedure.

We have generated a second set of harder SCP instances. Consider a grid graph with $x \times x$ vertices. The vertices of the graph correspond to the points in the plane with integer coordinates, and two vertices i and j are connected by two arcs, (i, j) and (j, i), whenever the corresponding points are at distance 1. Note that, unlike the first set of SCP instances, where the graph only

	Grid size x	Jobs n	Origins k	Dest. m	Gap0 (%)	# opt.	Time
Drayage100_a	100×100	100	50	75	0.00	5/5	0,1
Drayage100_b	$10^6 \times 10^6$	100	100	65	0.07	2/2	1,2
Drayage300_a	500×500	300	150	200	0.00	5/5	4,4
Drayage300_b	$10^6 \times 10^6$	300	300	200	0.002	2/2	45,3

Table 5: Results on drayage instances.

contained vertices incident with required arcs (jobs), here all the vertices of the grid are included in the graph. A set of n jobs is created by randomly choosing n origins and n destinations. If the distance between the origin and the destination is greater than a given threshold d, the job is discarded and a new one is generated. The distance metric used here is the Manhattan distance.

We have generated 50 instances on a grid of size 50×50 and 40 on a 100×100 grid. The number of jobs varies from 100 to 3000. Two possible values for the distance threshold d have been used, namely 5 and 8. Five instances have been generated with each different combination of parameters x, n, and d. We first tried to solve these instances directly on the original grid graphs, but the large number of non-required vertices and the fact that all the non-required arcs have cost 1, made them too difficult, because a huge number of connectivity inequalities is needed to guarantee the connectivity of the solution. Then we tried transforming the graph by removing all the non-required vertices and adding non-required arcs representing the shortest path between jobs. Although the number of such shortest paths is huge, due to the special structure of these grid graphs and the metric used to obtain the costs of the arcs, many of these shortest paths are redundant, and thus can be removed from the transformed graph. For example, in the instances with $x = 100$ and $n = 500$, the number of shortest paths is around 250000, while the transformed graph contains around 25000 non-required arcs.

Tables 6 and 7 show the results obtained on these sets of SCP instances. Note that all the instances with $x = 50$ have been solved to optimality in very short computing times. Also, 27 out of 40 instances with $x = 100$ have been solved to optimality within the time limit of one hour. For those instances that could not be solved optimally, column *Gap*(%) gives the average gap between the final lower bound and the best solution found.

Set	Jobs	d	# opt.	Gap0 (%)	Nodes	Time
Grid50_100_5	100	5	5/5	0,45	61	5,0
Grid50_100_8	100	8	5/5	0,10	6	1,4
Grid50_300_5	300	5	5/5	0,05	15,6	12,1
Grid50_300_8	300	8	5/5	0,00	1,8	3,7
Grid50_500_5	500	5	5/5	0,00	1	5,6
Grid50_500_8	500	8	5/5	0,00	0	2,8
Grid50_1000_5	1000	5	5/5	0,00	0	2,2
Grid50_1000_8	1000	8	5/5	0,00	0	2,1
Grid50_2000_5	2000	5	5/5	0,00	0	1,4
Grid50_2000_8	2000	8	5/5	0,00	0	1,6

Table 6: Results on SCP instances on a 50×50 grid graph.

Note that, in general, for a fixed number of jobs, the instances get easier when d takes a greater value. This can be explained by the different effort needed to guarantee the connectivity of the solution. Although the number of R-sets of two instances with the same characteristics but with $d = 5$ and $d = 8$ is similar, after solving the initial LP, the number of connected components induced by the LP solution is much smaller for the instance with $d = 8$ than for the one with $d = 5$. A graphical explanation of this behavior can be seen in Figure 7, where two

Set	Jobs	d	# opt.	Gap0 (%)	Gap (%)	Nodes	Time
Grid100_500_5	500	5	0/5	0,70	0,55	998,2	3600
Grid100_500_8	500	8	3/5	0,11	0,03	438,2	2033,0
Grid100_1000_5	1000	5	0/5	0,32	0,28	325,8	3600
Grid100_1000_8	1000	8	5/5	0,00	0,00	14,2	333,1
Grid100_2000_5	2000	5	4/5	0,01	0,00	76,6	1674,0
Grid100_2000_8	2000	8	5/5	0,00	0,00	0,6	110,2
Grid100_3000_5	3000	5	5/5	0,00	0,00	0	97,7
Grid100_3000_8	3000	8	5/5	0,00	0,00	0	78,1

Table 7: Results on SCP instances on a 100×100 grid graph.

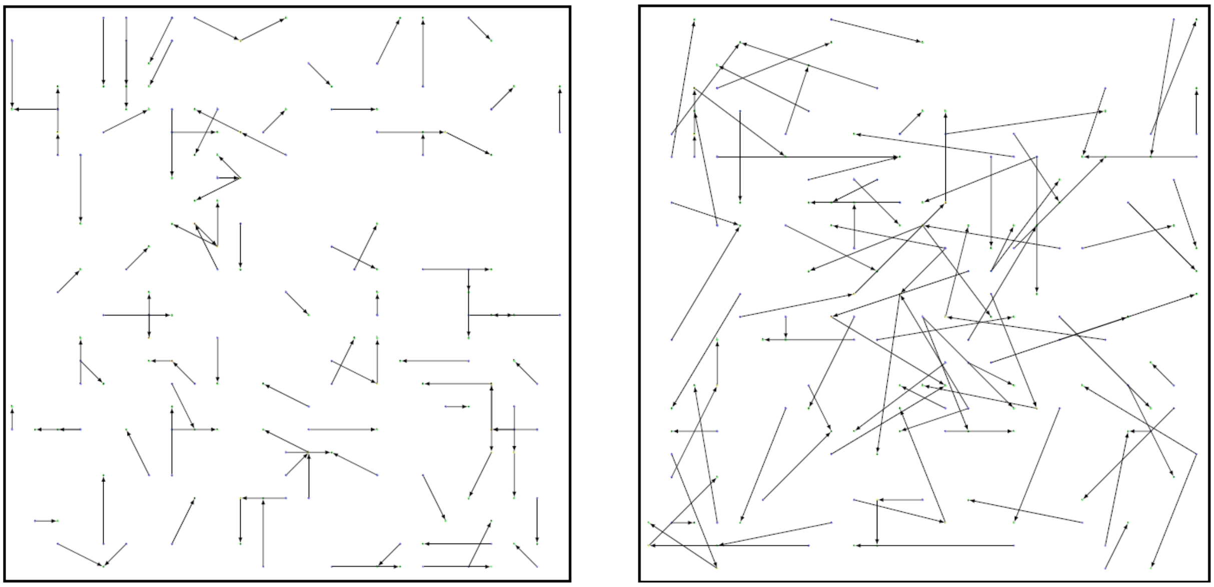

Figure 7: Two SCP instances with 100 jobs in a grid 25×25, $d = 3$ and $d = 8$.

SCP instances with 100 jobs in a grid 25 × 25 and different values of d, 3 and 8, are depicted. The number of R-sets is 74 and 71, respectively, while after solving the initial LP, the number of connected components of the solution reduces to 17 and 5, respectively. The first instance is solved to optimality after adding 22 connectivity cuts, while only five are needed for the second one. The same behavior can be observed when the number of jobs increases. For the SCP instance in Figure 8, with 500 jobs, the initial number of R-sets is smaller, 41, and the number of connected components of the first LP solution is one, which makes the instance trivial.

5 Conclusions

In this paper we have addressed the polyhedral description and the resolution of the Directed General Routing Problem (DGRP) and the Stacker Crane Problem (SCP). Unlike previous works on related arc routing problems, we have studied the DGRP and SCP on the original graph, instead of transforming it in order to remove all the non-required vertices. We have described some large families of facet-defining inequalities and implemented a branch-and-cut algorithm for these problems. We have carried out extensive computational experiments over different sets of DGRP and SCP instances. The results show that, while simplifying the graph

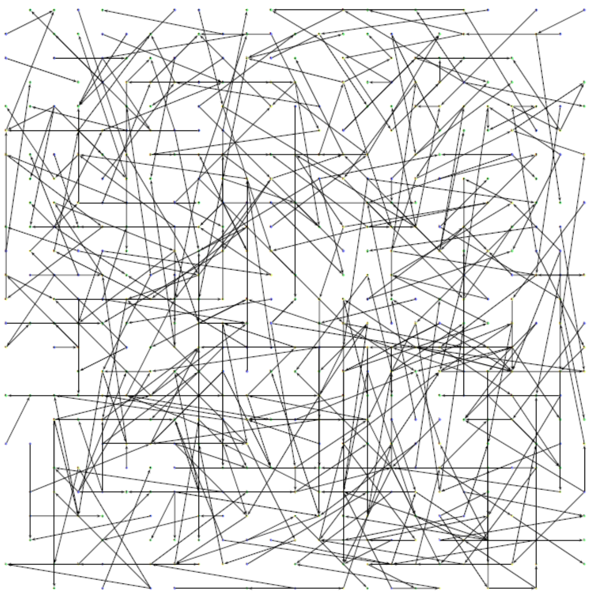

Figure 8: SCP instance with 500 jobs in a grid 25×25, $d = 8$.

can be useful in some types of instances, there are other situations in which working with the original graph seems to be the only successful way of solving the problem. Overall, we think that these results also prove that our algorithm is among the best solution procedures proposed for both problems.

Acknowledgements: Authors wish to thank the Ministerio de Economía y Competitividad (project MTM2012-36163-C06-02) of Spain and the Generalitat Valenciana (project GVPROMETEO2013-049) for their support.

References

[1] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15:1–31, 2007.

[2] M. Blais and G. Laporte. Exact solution of the generalized routing problem through graph transformations. *Journal of Operational Research Society*, 54:906–910, 2003.

[3] V. Campos and J.V. Savall. A computational study of several heuristics for the DRPP. *Computational Optimization and Applications*, 4:67–77, 1995.

[4] G. Carpaneto, M. Dell'Amico, and P. Toth. Exact solution of large scale, asymmetric traveling salesman problems. *ACM Transactions on Mathematial Software*, 21:395–409, 1995.

[5] N. Christofides, V. Campos, A. Corberán, and E. Mota. An algorithm for the rural postman problem on a directed graph. *Mathematical Programming Study*, 26:155–166, 1986.

[6] J. Cirasella, D.S. Johnson, L.A. McGeoch, and W. Zhang. Traveling salesman problem: Algorithms, instance generators, and tests. In A. L. Buchsbaum and J. Snoeyink, editors, *ALENEX 2001 Proceedings*, Lecture Notes in Computer Science 2153, pages 32–59. Springer, Berlin, 2001.

[7] Á. Corberán, G. Mejía, and J.M. Sanchis. New results on the mixed general routing problem. *Operations Research*, 53:363–376, 2005.

[8] Á. Corberán, I. Plana, and J.M. Sanchis. A branch & cut algorithm for the windy general routing problem and special cases. *Networks*, 49:245–257, 2007.

[9] Á. Corberán, A. Romero, and J.M. Sanchis. The mixed general routing polyhedron. *Mathematical Programming*, 96:103–137, 2003.

[10] H.A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems, part 2: The rural postman problem. *Operations Research*, 43:399–414, 1995.

[11] G.N. Frederickson, M.S. Hecht, and C.E. Kim. Approximation algorithms for some routing problems. In *17th Annual Symposium on Foundations of Computer Science*, pages 216–227, Houston, TX, 1976.

[12] G.N. Frederickson, M.S. Hecht, and C.E. Kim. Approximation algorithms for some routing problems. *SIAM Journal on Computing*, 7:178–193, 1978.

[13] R. Hassin and S. Khuller. z-approximations. *Journal of Algorithms*, 41:429–442, 2001.

[14] D.S. Johnson, G. Gutin, L.A. McGeoch, A. Yeo, W. Zhang, and A. Zverovitch. Experimental analysis of heuristics for the atsp. In G. Gutin and A.P. Punnen, editors, *The travelling salesman problem and its variations*, Combinatorial Optimization 12, pages 445–487. Kluwer Academic Publishers, Dordrecht, 2002.

[15] G. Laporte. Modeling and solving several classes of arc routing problems as traveling salesman problems. *Computers & Operations Research*, 24:1057–1061, 1997.

[16] D. Naddef and G. Rinaldi. The symmetric traveling salesman polytope and its graphical relaxation: Composition of valid inequalities. *Mathematical Programming*, 51:359–400, 1991.

[17] J.V. Savall. *Polyhedral results and approximate algorithms for the directed rural postman problem.* Ph.D. Thesis, University of Valencia, 1990.

[18] J. Srour. *Dissecting drayage: an examination of structure, information, and control in drayage operations.* Phd thesis, Erasmus Universiteit Rotterdam, 2010.

[19] J. Srour and S. van de Velde. Are stacker crane problems easy? A statistical study. *Computers & Operations Research*, 40, 2013.

[20] L. Zhang. Polynomial algorithms for the k-Chinese postman problem. In J. Van Leeuwen, editor, *Proceedings of the IFIP 12th World Computer Congress. Volume 1: Algorithms, Software, Architecture*, pages 430–435. Elsevier, Amsterdam, 1992.

[21] L. Zhang and W. Zheng. Genetic coding for solving both the stacker crane problem and its k-variant. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1061–1066. IEEE, Piscataway, NJ, 1995.