Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Computers & Operations Research 33 (2006) 3384–3402

computers & operations research

www.elsevier.com/locate/cor

A comparison of two different formulations for arc routing problems on mixed graphs

Angel Corberán[a,*], Enrique Mota[a], José M. Sanchis[b]

[a]*Departament d'Estadística i Investigació Operativa, Facultad de Matemáticas, Universitat de València, Dr. Moliner 50, 46100 Burjassot, Valencia, Spain*
[b]*Departamento de Matemática Aplicada, Universidad Politécnica de Valencia, Spain*

Available online 8 April 2005

Abstract

Arc routing problems on mixed graphs have been modelled in the literature either using just one variable per edge or associating to each edge two variables, each one representing its traversal in the corresponding direction. In this paper, and using the mixed general routing problem as an example, we compare theoretical and computationally both formulations as well as the lower bounds obtained from them using Linear Programming based methods. Extensive computational experiments, including some big and newly generated random instances, are presented.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Arc routing; Mixed Chinese postman problem; Mixed rural postman problem; Mixed general routing problem

1. Introduction

Arc routing problems consist of finding a minimum cost set of routes over all or some of the links of a graph satisfying some side constraints. See the book edited by Dror [1] for an excellent survey of the state of the art and applications of the field of arc routing. The basic problem, known as the Chinese postman problem (CPP), is that of determining a minimum cost closed walk traversing at least once each edge of a non directed graph. From this basic and simple problem, the research, motivated by real-world applications, has focused along the last two decades on increasingly more difficult and general problems.

* Corresponding author. Tel.: +34 6386 4306; fax: +34 6386 4735.
E-mail address: angel.corberan@uv.es (A. Corberán).

doi:10.1016/j.cor.2005.02.010

One step in this generalization is to consider an underlying graph consisting both of edges and arcs. Arcs (directed links) may model for example one-way streets, while two-way streets may be represented by edges (undirected links). When the basic CPP is defined on such a mixed graph, the problem is known as the mixed Chinese postman problem (MCPP), and it is *NP*-hard, as shown by Papadimitriou [2]. Ford and Fulkerson [3] were the first to present a characterization of those (Eulerian) mixed graphs that admit a closed walk traversing each link (edge or arc) exactly once. The MCPP was addressed by Christofides et al. [4], who proposed a Branch & Bound algorithm for its resolution. Using a similar formulation for a more general problem, the windy postman problem (WPP), Grötschel and Win [5] designed a cutting-plane algorithm capable of solving medium size MCPP instances to optimality. Some years later, Nobert and Picard [6], using a somewhat different formulation, presented another cutting-plane algorithm for this problem. This procedure is able to optimally solve instances of medium-large size. The MCPP is a special case of the mixed rural postman problem (MRPP), which consists of finding a minimum cost closed walk traversing a subset of the links of a mixed graph; these links are then the required ones, while the non-required links may be part of the solution. If, in a more general context, a subset of vertices requiring service is also considered, we have the mixed general routing problem (MGRP) that, although not strictly an arc routing problem, is one of the more general uncapacitated routing problems with a single vehicle and contains all the previous mentioned arc routing problems, as well as their undirected and directed versions, as special cases.

The MGRP has been recently studied in Corberán et al. [7] and Corberán et al. [8]. In the second paper a cutting-plane algorithm based on most of the known valid and facet-defining inequalities of the MGRP polyhedron is presented. This MGRP study starts from a formulation in which only one variable is associated to each edge. As the formulation proposed by Nobert and Picard [6] for the MCPP, it is based on the characterization of an Eulerian mixed graph given by Ford and Fulkerson [3]. But some routing problems on mixed graphs have also been modelled using two variables associated to the same edge (see Kappauf and Koehler [9], Christofides et al. [4], Grötschel and Win [5] and Ralphs [10]), each variable representing its traversal in the corresponding direction. In fact, during the revision of [7], an anonymous referee argued that formulating the MGRP using two variables for each edge would produce a stronger LP relaxation. To convince him/her, authors did some computational experiments that showed that the lower bounds obtained from the formulation using only one variable per edge were better than those obtained with the other one. But later on an error was detected in the coding of the cutting-plane algorithm based on the two variables per edge formulation. Once the mistake was corrected and the computational experiments repeated, the results were then not so conclusive. Therefore, from the above discussion, we think that this important point deserves to be studied in depth. This is the purpose of this paper.

We will compare these two alternative formulations to arc routing problems on mixed graphs choosing a very general problem, the MGRP, as an example. In Section 2 we define precisely the problem, introducing the notation to be used along the following sections and presenting the two different integer formulations of the MGRP, which are shown to be equivalent. In Section 3, and for each formulation, we obtain initial lower bounds from linear programming based methods. Although the initial LP relaxations are not equivalent, we obtain equivalent linear formulations once every valid inequality from the classes that are known to be separated in polynomial time is also considered. Bounds obtained from both 'polynomial' linear relaxations can be further improved by adding other valid inequalities as they are (heuristically) detected. Section 4 presents the computational experiments in which both formulations are compared over an extensive set of newly generated instances. The last section includes a brief summary and the conclusions of the research.

2. Problem definition and formulations

Given a strongly connected mixed graph $G=(V, E, A)$ with vertex set V, edge set E, arc set A, a cost c_e for each link $e \in E \cup A$, a set $E_{\mathrm{R}} \subseteq E$ of *required edges*, a set $A_{\mathrm{R}} \subseteq A$ of *required arcs* and a set $V_{\mathrm{R}} \subseteq V$ of *required vertices*, the MGRP is the problem of finding a minimum cost closed walk (a *tour*) passing through each required link $e \in E_{\mathrm{R}} \cup A_{\mathrm{R}}$ and through each $i \in V_{\mathrm{R}}$ at least once.

Note that if $i \in V$ is a vertex incident to any required link $e \in E_{\mathrm{R}} \cup A_{\mathrm{R}}$, the condition on the tour passing through link e contains the condition of visiting vertex i. Therefore, in what follows, we will assume that V_{R} contains the set of vertices incident to the required edges. Furthermore, as it is usual when the subgraph induced by the required vertices and links is not connected, we first transform the original graph in order to simplify both the problem structure and the formulation. This transformation is done in a similar way to that of Christofides et al. [11] for the undirected RPP (see Eiselt et al. [12], for an illustration of the procedure). The graph resulting from this transformation is such that all its vertices are required ones as well as its edges, i.e., $V=V_{\mathrm{R}}$ and $E=E_{\mathrm{R}}$. Then, we will assume that the MGRP is defined on a strongly connected mixed graph $G=(V, E, A):=(V_{\mathrm{R}}, E_{\mathrm{R}}, A_{\mathrm{R}} \cup A_{\mathrm{NR}})$.

We will use the same notation as in [7] and [8]. Let $G^{\mathrm{R}}=G(V, E, A_{\mathrm{R}})$ be the graph obtained by deleting in G all the non required arcs A_{NR}. In general, graph G^{R} is not connected. Let p be the number of connected components of G^{R} and let $V_1, V_2, \ldots, V_p$ be the vertex sets corresponding to the p connected components of G^{R}, which will be called R-sets, with $V_1 \cup \cdots \cup V_p = V$. We will represent by $C_i=G(V_i),\ i=1, \ldots, p$, the subgraphs of G induced by the R-sets and they will be referred to as R-connected components. Notice that every isolated required vertex is a R-connected component of G. Given two disjoint sets of vertices $S_1,\ S_2 \subset V$ and a set $S \subset V$, we denote

$$
\begin{aligned}
&(S_1 : S_2)=\{(i, j) \in E \cup A : i \in S_1,\ \ j \in S_2 \ or \ i \in S_2,\ \ j \in S_1\},\\
&\delta(S)=(S : V \backslash S) \text{ (called link cut-set of } G \text{ defined by } S),\\
&A^{+}(S)=A(S : V \backslash S),\\
&A^{-}(S)=A(V \backslash S : S),\\
&E(S)=E(S : V \backslash S).
\end{aligned}
$$

The following sets are defined in a similar way referring to required links only and to non-required arcs only: $A_{\mathrm{R}}^{+}(S)$, $A_{\mathrm{NR}}^{-}(S)$, $\delta_{\mathrm{R}}(S)$, etc. Given $x \in \mathbb{R}^{|E \cup A|}$ and given $T \subset E \cup A$, $x(T)$ denotes $\sum_{e \in T} x_e$.

A *tour for the* MGRP is a family $\mathscr{F}$ of links of G such that the graph $(V, E^{\mathscr{F}} \cup A^{\mathscr{F}})$ contains all the required links and vertices and a closed walk traversing each copy of each link exactly once (i.e., is an Eulerian graph), where $E^{\mathscr{F}} \cup A^{\mathscr{F}}$ is obtained by considering each copy of a link in $\mathscr{F}$ as a different element.

Applying the necessary and sufficient conditions for a connected mixed graph to be Eulerian ([3]), we can state that $\mathscr{F}$ is a tour for the MGRP when the following conditions are satisfied:

- $\mathscr{F}$ contains all the required links,
- graph $(V, E^{\mathscr{F}} \cup A^{\mathscr{F}})$ is connected,
- graph $(V, E^{\mathscr{F}} \cup A^{\mathscr{F}})$ is even, i.e., every vertex is incident to an even number of links,
- graph $(V, E^{\mathscr{F}} \cup A^{\mathscr{F}})$ is balanced, i.e., for every $S \subset V$, the difference between the number of arcs in $A^{+}(S)$ (leaving S) and the number of arcs in $A^{-}(S)$ (entering S) is less than or equal to the number of edges in $E(S)$ (*balanced-set condition* for set S).

The family of links obtained from any tour for the MGRP by deleting one copy of every link in $E \cup A_{\mathrm{R}}$ is called a *semitour for the* MGRP. Although in [7] and [8] the MGRP is formulated with respect to semitours, in this paper and for the sake of simplicity, we will always consider MGRP tours. Associated to each MGRP tour there is an integer *incidence vector* $y = (y_e : e \in E \cup A) \in \mathbb{R}^{|E \cup A|}$, where y_e denotes the number of times a link $e \in E \cup A$ appears in the tour. If it is necessary to make an explicit reference to the direction in which a required link is traversed, we will use y_{ij} instead of y_e. A vertex $v \in V$ will be called *R-odd* if it is incident to an odd number of required links, otherwise it will be called *R-even*. Note that every isolated required vertex is *R*-even.

It is easy to see that the set of tours for the MGRP is then the set of vectors $y \in \mathbb{R}^{|E \cup A|}$ satisfying

$$y_{ij} \geqslant 0 \textit{ and integer}, \quad \forall (i,j) \in A_{\mathrm{NR}}, \tag{1}$$

$$y_{ij} \geqslant 1 \textit{ and integer}, \quad \forall (i,j) \in A_{\mathrm{R}}, \tag{2}$$

$$y_e \geqslant 1 \textit{ and integer}, \quad \forall e \in E_{\mathrm{R}}, \tag{3}$$

$$y(\delta(\{i\})) \equiv 0 \bmod 2, \quad \forall i \in V : i \textit{ is R-even}, \tag{4}$$

$$y(\delta(\{i\})) \equiv 0 \bmod 2, \quad \forall i \in V : i \textit{ is R-odd}, \tag{5}$$

$$y(A^+(S)) \geqslant 1, \quad \forall S = \bigcup_{k \in Q} V_k, \ Q \subset \{1, \ldots, p\}, \tag{6}$$

$$y(A^+(S)) - y(A^-(S)) \leqslant y(E(S)), \quad \forall S \subset V, \tag{7}$$

where (2) and (3) imply that all the required links are in the solution, (4) and (5) and (6) assure that the resulting graph will be even and connected, respectively, and (7) that the balanced-set conditions will be satisfied. This formulation of the MGRP will be denoted by F1 and (with respect to semitours) was the one used in [7] and [8].

Consider now a set $S \subset V$ such that $E(S) = \emptyset$. Then the balanced-set condition corresponding to S and $\bar{S} = V \backslash S$ implies the so called *symmetry equation* $y(A^+(S)) = y(A^-(S))$. Hence, the above formulation includes an equation associated with each set $S \subset V$ with $E(S) = \emptyset$. Although most of these equations will be linearly dependent, if we consider the q subgraphs of G induced by the required edges, it can be shown that any $q - 1$ of the corresponding symmetry equations are linearly independent. In [7] these equations

$$y(A^+(K_i)) = y(A^-(K_i)), \quad i = 1, 2, \ldots, q \tag{8}$$

are referred to as *the system equations*, where $K_1, K_2, \ldots, K_q$ denote the sets of vertices of the connected components of the graph (V, E). Note that some sets K_i could consist of a single vertex, that each set K_i is contained in a set V_j and that $E(V_j) = \emptyset, \forall j$.

Alternatively, if we choose the sufficient conditions for a mixed graph to be Eulerian ([3,13]), $\mathscr{F}$ is a tour for the MGRP if

- $\mathscr{F}$ contains all the required links,
- graph $(V, E^{\mathscr{F}} \cup A^{\mathscr{F}})$ is connected,
- graph $(V, E^{\mathscr{F}} \cup A^{\mathscr{F}})$ is even,

- graph $(V, A^{\mathscr{F}})$ is symmetric, i.e., the number of arcs entering every vertex is equal to the number of arcs leaving it.

Note that if every vertex is symmetric, the number of arcs entering any $S \subset V$ should also be equal to the number of arcs leaving it.

The above conditions can be used to formulate the problem by associating two variables x_{ij} and x_{ji} to each edge $e=(i, j)$, representing the number of times edge e is traversed in the corresponding direction. Then, the set of tours for the MGRP, according to this second formulation, called F2 in what follows, is the set of vectors $x \in \mathbb{R}^{2|E|+|A|}$ satisfying

$$x_{ij} \geqslant 0 \textit{ and integer}, \quad \forall (i, j) \in A_{\mathrm{NR}}, \tag{9}$$

$$x_{ij} \geqslant 1 \textit{ and integer}, \quad \forall (i, j) \in A_{\mathrm{R}}, \tag{10}$$

$$x_{ij}, x_{ji} \geqslant 0 \textit{ and integer}, \quad \forall e=(i, j) \in E_{\mathrm{R}}, \tag{11}$$

$$x_{ij} + x_{ji} \geqslant 1, \quad \forall e=(i, j) \in E_{\mathrm{R}}, \tag{12}$$

$$x(A^{+}(S)) \geqslant 1, \quad \forall S = \bigcup_{k \in Q} V_k, \ \ Q \subset \{1, \ldots, p\}, \tag{13}$$

$$x(A^{+}(i)) + \sum_{j:(i,j)\in E_{\mathrm{R}}} x_{ij} = x(A^{-}(i)) + \sum_{j:(i,j)\in E_{\mathrm{R}}} x_{ji}, \quad \forall i \in V, \tag{14}$$

where (10) and (12) imply that all the required links are in the solution, (13) assure that the resulting graph will be connected and (14) that the symmetry conditions will be satisfied. Note that integrality and symmetry conditions imply that the resulting graph will be even.

We now show that formulations F1 and F2 are equivalent. Let x be a solution to F2 and define $y_e = x_{ij} + x_{ji}$, $\forall e=(i, j) \in E_{\mathrm{R}}$ and $y_{ij} = x_{ij}$, $\forall (i, j) \in A$. Then, constraints (9)–(12) imply constraints (1)–(3). Furthermore, since x induces a connected and symmetric graph, y defines an even and connected graph. And given that every vertex is symmetric with respect to x, a symmetry equation holds for each cut $(S : V\backslash S)$ and, therefore, the balanced-set condition with respect to y is satisfied for each S. Thus, from any feasible solution to F2 we can obtain a feasible solution to F1 with the same cost. Conversely, consider now a solution to F1: it induces an even, connected and balanced mixed graph. Edmonds and Johnson [13] proposed a method of assigning a direction to the edges of an even and balanced mixed graph in order to obtain, with the same cost, a symmetric directed graph. Thus, from any feasible solution to F1 we can obtain a feasible solution to F2 with the same cost.

3. Lower bounds from linear programming based methods

Once we have proved that both (integer) formulations are equivalent, we are now interested in comparing them in order to obtain lower bounds for the MGRP using linear programming based methods. To do this the *integrality* conditions in F1 and F2 must be removed. Furthermore, constraints (4) and (5) in F1 are not linear and there is not a linear expression of them without using integer variables. Hence, these constraints should also be removed. However, it is easy to see that for each R-odd vertex i, $y(\delta(i)) \geqslant |\delta_{\mathrm{R}}(i)| + 1$ must hold. In a similar way, given an R-odd cutset $\delta(S)$ (a cutset containing an odd number of required links), the following inequalities, called *R-odd cut inequalities*, must be satisfied by any solution y to F1 (see [7])

$$y(\delta(S)) \geqslant |\delta_{\mathrm{R}}(S)| + 1, \quad \forall \delta(S), \quad R\text{-odd cutset.} \tag{15}$$

On the other hand, note that if integrality conditions are satisfied, symmetry inequalities (14) imply that all the vertices are even. But, once the integrality conditions have been removed from F2, symmetry conditions alone do not longer guarantee the evenness of the vertices. However, constraints (15) should also be satisfied by any solution x to F2.

Therefore, we will call $\mathrm{LP_{F1}}$ the following linear relaxation of F1:

$$y_{ij} \geqslant 0, \quad \forall (i,j) \in A_{\mathrm{NR}}, \tag{16}$$

$$y_{ij} \geqslant 1, \quad \forall (i,j) \in A_{\mathrm{R}}, \tag{17}$$

$$y_e \geqslant 1, \quad \forall e \in E_{\mathrm{R}}, \tag{18}$$

$$y(A^+(S)) \geqslant 1, \quad \forall S = \bigcup_{k \in Q} V_k, \ \ Q \subset \{1, \ldots, p\}, \tag{19}$$

$$y(A^+(S)) - y(A^-(S)) \leqslant y(E(S)), \quad \forall S \subset V, \tag{20}$$

$$y(\delta(S)) \geqslant |\delta_{\mathrm{R}}(S)| + 1, \quad \forall \delta(S), \ \ R\text{-odd cutset} \tag{21}$$

and $\mathrm{LP_{F2}}$ the following linear relaxation of F2:

$$x_{ij} \geqslant 0, \quad \forall (i,j) \in A_{\mathrm{NR}}, \tag{22}$$

$$x_{ij} \geqslant 1, \quad \forall (i,j) \in A_{\mathrm{R}}, \tag{23}$$

$$x_{ij}, x_{ji} \geqslant 0, \quad \forall e = (i,j) \in E_{\mathrm{R}}, \tag{24}$$

$$x_{ij} + x_{ji} \geqslant 1, \quad \forall e = (i,j) \in E_{\mathrm{R}}, \tag{25}$$

$$x(A^+(S)) \geqslant 1, \quad \forall S = \bigcup_{k \in Q} V_k, \ \ Q \subset \{1, \ldots, p\}, \tag{26}$$

$$x(\delta(S)) \geqslant |\delta_{\mathrm{R}}(S)| + 1, \quad \forall \delta(S), \ \ R\text{-odd cutset.} \tag{27}$$

$$x(A^+(i)) + \sum_{j:(i,j) \in E_{\mathrm{R}}} x_{ij} = x(A^-(i)) + \sum_{j:(i,j) \in E_{\mathrm{R}}} x_{ji}, \ \ \forall i \in V. \tag{28}$$

3.1. Initial lower bounds

Note that constraints (19)–(21) in $\mathrm{LP_{F1}}$ and (26)–(27) in $\mathrm{LP_{F2}}$ are exponential in number. Therefore, only a subset of them can be explicitly added to the LP's to be solved. As in [7] and [8], we define the following initial LP relaxation, $\mathrm{LP0_{F1}}$, corresponding to $\mathrm{LP_{F1}}$ containing:

- the system Eq. (8),
- constraints (17) and (18),

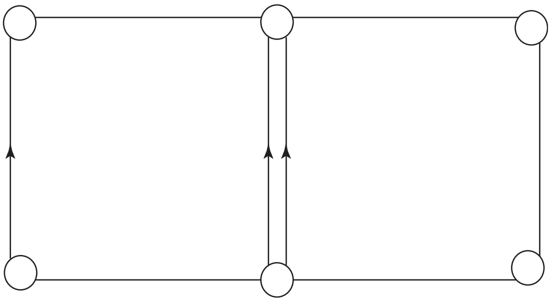

Fig. 1. Feasible solution to $\mathrm{LP0_{F1}}$.

- one connectivity inequality (19) for each R-set,
- one balanced-set inequality (20) for each 'unbalanced' vertex,
- one R-odd cut inequality (21) for each R-odd 'balanced' vertex,

where an unbalanced vertex i is a vertex for which $|E(i)| < |A_{\mathrm{R}}^{+}(i)| - |A_{\mathrm{R}}^{-}(i)|$ or $|E(i)| < |A_{\mathrm{R}}^{-}(i)| - |A_{\mathrm{R}}^{+}(i)|$.

In a similar way we define $\mathrm{LP0_{F2}}$ (corresponding to $\mathrm{LP_{F2}}$) as the LP containing:

- constraints (23) and (25),
- one connectivity inequality (26) for each R-set,
- one R-odd cut inequality (27) for each R-odd 'balanced' vertex,
- one symmetry (28) for each vertex.

In what follows, we show that $\mathrm{LP0_{F2}}$ is stronger than $\mathrm{LP0_{F1}}$. Consider a feasible solution x to $\mathrm{LP0_{F2}}$ and define again $y_e = x_{ij} + x_{ji}$, $\forall e = (i, j) \in E_{\mathrm{R}}$ and $y_{ij} = x_{ij}$, $\forall (i, j) \in A$. If x satisfies the symmetry constraints (28) for every vertex, y satisfies all the system equations and also the balanced-set constraints (20) for each vertex. Then, from any feasible solution to $\mathrm{LP0_{F2}}$ we obtain a feasible solution to $\mathrm{LP0_{F1}}$ with the same cost. However, the converse is not true, as the following example shows. The graph in Fig. 1, in which all the links are required, represents a feasible solution to $\mathrm{LP0_{F1}}$ (although the balanced-set conditions are not satisfied for every subset S of vertices). However, there is no way of directing the edges so as, with the same cost, obtaining a symmetric graph. Therefore, the lower bound obtained using $\mathrm{LP0_{F1}}$ cannot be greater than the lower bound obtained using $\mathrm{LP0_{F2}}$. This fact is reflected in column initial lower bound (LB0) in the tables in Section 4.

3.2. Lower bounds in polynomial time

Since there is, as already mentioned, an exponential number of inequalities in some of the families defining $\mathrm{LP_{F1}}$ and $\mathrm{LP_{F2}}$, not all of them can be explicitly included in a LP to be solved. One alternative in order to solve these programs is to use the following iterative process, called *cutting-plane algorithm*. To begin with, compute the solution of an LP consisting of a small subset of inequalities (in our case, those in $\mathrm{LP0_{F1}}$ and $\mathrm{LP0_{F2}}$). Then, look for inequalities, not in the current LP, that are violated by its optimal solution. If one or more violated inequalities are detected, add them to the current LP and solve it again; otherwise, terminate. This process needs to solve the 'separation problem', i.e., the identification of those inequalities that are violated by the current LP solution. This problem is solved in practice using a different algorithm for each class of valid inequalities. An *exact separation algorithm* for a given class of inequalities is a routine that, having an LP solution as input, outputs one or more violated inequalities in that class, whenever there are violated inequalities. A *heuristic separation algorithm* works in a similar way, but it may fail to detect violated inequalities in the class.

The separation problems associated with connectivity and R-odd cut inequalities are solvable in polynomial time by means of max-flow calculations and the Padberg and Rao [14] procedure to find minimum odd cut-sets. Nobert and Picard [6] showed that identifying violated balanced-set inequalities for the MCPP can also be solved in polynomial time. This result is easily extended to the MGRP. Therefore, since the separation and the optimization problems are equivalent (see Grötschel et al. [15]), $\mathrm{LP}_{\mathrm{F1}}$ and $\mathrm{LP}_{\mathrm{F2}}$ can be solved in polynomial time (assuming that the ellipsoid method for linear programming is used).

In what follows, we will show that these two linear problems are equivalent. As before, it is easy to see that the set of feasible solutions to $\mathrm{LP}_{\mathrm{F2}}$ is included in that of $\mathrm{LP}_{\mathrm{F1}}$. Consider now a (possibly fractional) feasible solution y to $\mathrm{LP}_{\mathrm{F1}}$. Let G_y be the weighted graph with vertex set V, link set $\{e \in E \cup A : y_e > 0\}$ and link-weights y_e. The question is how to assign a direction to the edges of G_y in order to obtain a directed graph satisfying the symmetry equations. To do this, we first solve a flow problem on an undirected graph $H = (V_H, E_H)$, where $V_H = V \cup \{0, n+1\}$ (0 and $n+1$ are two extra vertices). The set of edges E_H includes all the edges in E, with an associated capacity y_e, plus one edge from vertex 0 to every vertex $i \in V$ with capacity $y_{0i} = \max\{d(i), 0\}$, where $d(i) = y(A^-(i)) - y(A^+(i))$, and one edge from each $i \in V$ to vertex $n+1$ with capacity $y_{i,n+1} = \max\{-d(i), 0\}$. Let $P = \sum_{d(i)>0} d(i) = \sum_{d(i)<0} - d(i)$.

Compute a maximum flow in H from vertex 0 to vertex $n+1$. Since y satisfies all the balanced-set inequalities, the flow equals P (otherwise, as it is shown in [16], a balanced-set inequality violated by y would exist). Hence, this implies that it is possible to satisfy all the supplies ($d(i) > 0$) and demands ($d(i) < 0$) of the vertices in G_y, computed with respect to its arcs, using only its edges.

Now, let $G(A)$ be the weighted graph induced by the arcs in G_y. From the optimal solution f to the above flow problem,

- if the optimal flow through a given edge e is 0, add to $G(A)$ two arcs, in opposite directions, each one with weight $y_e/2$,
- if the optimal flow through an edge equals its capacity, add to $G(A)$ one arc with weight y_e in the direction given by the flow,
- if the optimal flow f_e through an edge $e = (i, j)$ is less than its capacity, add to $G(A)$ one arc in the direction given by the flow, with weight f_e. Add also two more arcs (i, j) and (j, i) with weights equal to $(y_e - f_e)/2$.

The resulting graph is symmetric and has the same cost as G_y. Note that the above process is equivalent to constructing a feasible solution x to $\mathrm{LP}_{\mathrm{F2}}$ by defining $x_{ij} = y_{ij}$ for all arcs (i, j) in A and $x_{ij} = (y_e + f_e)/2$ and $x_{ji} = (y_e - f_e)/2$ for every edge $e = (i, j)$, if (i, j) is the direction given by the flow. Therefore, from a feasible solution y to $\mathrm{LP}_{\mathrm{F1}}$ we have obtained a feasible solution x to $\mathrm{LP}_{\mathrm{F2}}$ with the same cost.

Hence, $\mathrm{LP}_{\mathrm{F1}}$ and $\mathrm{LP}_{\mathrm{F2}}$ are equivalent and the bounds obtained from them (column LBpol in tables) are equal, as it can also be seen from the computational experiments.

3.3. Improving the bounds

In the previous subsection, the cutting-plane algorithm uses only the inequalities that appear in $\mathrm{LP}_{\mathrm{F1}}$ and $\mathrm{LP}_{\mathrm{F2}}$. Furthermore, these inequalities belong to classes that can be separated in polynomial time. However, the cutting-plane methodology allows the addition of any other valid inequality (one that is satisfied by all the feasible solutions) in order to improve the bound. The polyhedron associated to the MGRP solutions defined by F1, MGRP(G), has been studied in [8] and [7]. Besides connectivity, R-odd cut and balanced-set inequalities, other families of facet-inducing inequalities are also described: (standard) *K-C*, K-C_{02}, (standard) *Honeycomb* and *Honeycomb*$_{02}$.

Since no exact polynomial algorithm is known for the separation problems of all these latter inequalities, in the cutting-plane procedure described in [8], heuristic procedures were used to find violated inequalities of these classes. This cutting-plane algorithm has also been used here in order to get a better lower bound. Hence, in what follows we will represent by $\mathrm{LB}_{\mathrm{F1}}$ the lower bound obtained using the cutting-plane procedure described in [8].

Although the polyhedron associated to the MGRP solutions defined by F2 has not been explicitly studied, some valid inequalities can be easily derived from those known for polyhedron MGRP(G). Consider a feasible solution x to F2. As before, by adding the two x-variables corresponding to each edge $e = (i, j)$, we obtain a solution y to F1. Solution y should satisfy all the valid inequalities known for MGRP(G). Let $F(y) = \sum c_e y_e \geqslant b$ be such an inequality. It is easy to see that the inequality $F(x) = \sum c_e(x_{ij} + x_{ji}) \geqslant b$ is valid for x. Then, standard *K-C*, K-C_{02}, standard Honeycomb and Honeycomb$_{02}$ inequalities can be slightly modified in order to be valid inequalities for the MGRP solutions defined by F2. The corresponding separation procedures have been also modified and used to obtain a better lower bound $\mathrm{LB}_{\mathrm{F2}}$ by means of the cutting-plane algorithm based on formulation F2.

It is not known whether the inequalities mentioned in this section can or cannot be separated in polynomial time, but our guess is that these problems are $\mathcal{NP}$-hard. The use of heuristic algorithms to find violated inequalities of these classes implies that, although from a theoretical point of view bounds $\mathrm{LB}_{\mathrm{F1}}$ and $\mathrm{LB}_{\mathrm{F2}}$ could be equal, in practice they can provide different values. The computational results shown in the next section confirm this fact.

4. Computational testing

In this section we present the computational experiments conducted and the results obtained using the cutting-plane algorithms described in the previous sections.

Formulation F1 has been presented up to now with respect to tours to be more easily compared with formulation F2. However, the equivalent formulation with respect to semitours presents some computational advantages, since it handles implicitly constraints (17) and (18) and some inequalities can be written in a nicer way. Hence, this was the formulation used in the computational experiments.

4.1. The instances

In order to study in depth the differences between the two formulations, we have generated several sets of instances corresponding to some of the more important arc routing problems on mixed graphs, the MCPP, MRPP and MGRP. Let us begin with a description of the procedure used to generate these sets of instances.

On one hand, we have used graphs modelling the real street network of the Spanish towns of Albaida, Madrigueras and Aldaya. Graphs of Albaida and Madrigueras are undirected graphs having 116 vertices and 174 edges and 196 vertices and 316 edges, respectively. Although the Aldaya real street network corresponds to a mixed graph having 214 vertices and 224 edges and 127 arcs, here it will be considered as an undirected graph with 351 edges.

From these graphs, we first generate MCPP instances in the following way. All the edges in E are read in sequential order and some of them are transformed into an arc with a given probability p. If edge (i, j) is selected to be transformed into an arc, an orientation (i, j) or (j, i) is assigned with probability 0.5. Finally, if the number of strongly connected components is greater than one, some arcs are transformed back into edges to obtain a strongly connected graph. From each graph (Albaida, Madrigueras and Aldaya) and for each value of $p \in \{0.3, 0.5, 0.7\}$ we have generated three different instances. As an example, albaC3A, albaC3B and albaC3C are the three corresponding to the Albaida graph and $p = 0.3$ ('C' means Chinese). Then, we have 27 instances for the MCPP.

A similar process is used to generate instances for the MRPP and for the MGRP. Again, each edge is transformed into an arc with probability $p \in \{0.3, 0.5, 0.7\}$ and labelled as 'required' with probability $q \in \{0.3, 0.5, 0.7\}$. At this point we have an instance for the MGRP if we consider all the vertices as 'required'. In order to obtain an instance for the MRPP in which all the vertices are incident with required links, we iteratively select a link e with one end-point that is not incident with a 'required' link and label it as 'required' with the same probability q. This second step is repeated until all the vertices in V are incident with, at least, one 'required' link. From each graph (Albaida, Madrigueras and Aldaya), and for each value of parameters $p \in \{0.3, 0.5, 0.7\}$ and $q \in \{0.3, 0.5, 0.7\}$, we have generated 27 instances, named albaR33 to aldaR77 ('R' means Rural), for the MRPP and 27 other instances, named albaG33 to aldaG77 ('G' means General), for the MGRP.

On the other hand, in order to test the bounds on larger MRPP and MGRP instances, we decided to generate new random graphs. We first randomly selected 500 vertices in a 1000 × 1000 square. Edges were then generated as pairs of vertices i and j, with costs defined by $c_{ij} = \lfloor b_{ij} + \frac{1}{2} \rfloor$, where b_{ij} are the Euclidean distances. This is the same cost function proposed in the TSPLIB (see [17]). In order to obtain graphs with a structure similar to that of real street networks, we did not generate edges completely at random. Indeed, we proceeded as follows. Let d (degree) be a parameter. Add an edge from each vertex i to its d nearest vertices. Then remove those edges (i, j) for which $c_{ij} \geqslant 0.98(c_{ik} + c_{kj})$, for some (i, k) and (k, j). Some of the remaining edges are then considered as 'required' with a given probability q. If the resulting graph is not connected, edges in the d shortest trees spanning the connected components are also added and labelled as 'non required'. At this point and as before, if we consider all the vertices as 'required', we have an undirected GRP instance. Otherwise, to obtain an undirected RPP instance, the simplification procedure mentioned in Section 2 is applied. In both cases, from the undirected graphs, mixed instances are obtained as described above, where edges are transformed into arcs using a given probability p. By combining values {4, 5} for parameter d and values {0.25, 0.50, 0.75} for parameters q and p, we have generated 18 MRPP instances, named R422 (meaning 'Rural', $d = 4$, $q = 0.25$, $p = 0.25$) to R577 and 18 MGRP instances, named G422–G577.

4.2. Computational results

The algorithms were coded in C and ran on a PC with a 2000 MHz Pentium IV processor. Tables 1–11 show the computational results obtained for the instances described above.

Table 1
MCPP instances from Albaida graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	Dev (%)
AlbaC3A F1	1	116	174	14800	16380*	4.4	
AlbaC3A F2				15575	16380*	5.8	
AlbaC3B F1				14990	17256*	6.6	
AlbaC3B F2				16306	17256*	6.8	
AlbaC3C F1				15310	19518*	5.3	
AlbaC3C F2				18834	19518*	4.1	
AlbaC5A F1	1	116	174	19336	24568*	5.3	
AlbaC5A F2				23882	24568*	1.6	
AlbaC5B F1				15754	18216*	2.1	
AlbaC5B F2				17444	18216*	1.9	
AlbaC5C F1				17245	21340*	2.1	
AlbaC5C F2				20914	21340*	1.9	
AlbaC7A F1	1	116	174	16715	19328*	1.4	
AlbaC7A F2				18906	19328*	1.3	
AlbaC7B F1				20430	25462*	1.1	
AlbaC7B F2				25076	25462*	0.7	
AlbaC7C F1				16183	20560*	3.1	
AlbaC7C F2				20047	20560*	1.9	

Table 2
MCPP instances from Madrigueras graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	Dev (%)
MadrC3A F1	1	196	316	28165	31610*	19	
MadrC3A F2				31110	31610*	1.8	
MadrC3B F1				28245	31115*	15	
MadrC3B F2				30580	31115*	5.4	
MadrC3C F1				29385	35140*	14	
MadrC3C F2				34730	35140*	3.2	
MadrC5A F1	1	196	316	31445	38355*	9.1	
MadrC5A F2				37820	38355*	1.8	
MadrC5B F1				32605	41085*	6.7	
MadrC5B F2				40975	41085*	0.9	
MadrC5C F1				31780	39730*	9.5	
MadrC5C F2				39435	39730*	0.9	
MadrC7A F1	1	196	316	40505	52735*	3.3	
MadrC7A F2				52572.5	52735*	0.5	
MadrC7B F1				37120	44540*	4.3	
MadrC7B F2				44360	44540*	0.7	
MadrC7C F1				34445	41210*	5.0	
MadrC7C F2				40790	41210*	3.4	

Table 3
MCPP instances from Aldaya graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	Dev (%)
AldaC3A F1	1	214	351	41966	44422*	10	
AldaC3A F2				43230	44422*	6.1	
AldaC3B F1				42508	47480*	13	
AldaC3B F2				46623.5	47480*	3.6	
AldaC3C F1				42101	47344*	9.8	
AldaC3C F2				46824.5	47344*	2.1	
AldaC5A F1	1	214	351	45680	54143*	6.9	
AldaC5A F2				53508	54143*	1.1	
AldaC5B F1				52108	64108*	14	
AldaC5B F2				63708	64108*	1.5	
AldaC5C F1				48595	55999*	5.7	
AldaC5C F2				55280.5	55999*	0.9	
AldaC7A F1	1	214	351	55272	64479*	9.5	
AldaC7A F2				64197	64479*	3.1	
AldaC7B F1				66289	77849*	6.0	
AldaC7B F2				77591	77849*	1.1	
AldaC7C F1				53780	67013*	6.5	
AldaC7C F2				66681.5	67013*	1.6	

Table 4
MRPP instances from Albaida graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
AlbaR33 F1	24	116	174	10457	11300	1.8	11300	2.5	0.21
AlbaR33 F2	24			10482	11300	1.5	11300	3.2	0.21
AlbaR35 F1	30			13581.5	15710	0.6	15749*	1.3	
AlbaR35 F2	30			15026	15710	0.4	15734	0.7	0.09
AlbaR37 F1	24			11272	12982	1.0	13000*	1.0	
AlbaR37 F2	24			11802	12982	0.7	13000*	1.0	
AlbaR53 F1	19	116	174	11489	12608*	1.4	12608*	1.1	
AlbaR53 F2	19			11865	12608*	0.8	12608*	1.2	
AlbaR55 F1	13			11038	12520*	1.0	12520*	0.9	
AlbaR55 F2	13			11958	12520*	0.8	12520*	1.1	
AlbaR57 F1	12			13369	15619*	2.6	15619*	2.7	
AlbaR57 F2	12			14871	15619*	2.1	15619*	3.1	
AlbaR73 F1	8	116	174	13521	15301*	1.9	15301*	1.8	
AlbaR73 F2	8			14875	15301*	0.9	15301*	1.3	
AlbaR75 F1	2			13897	16550	2.2	16550	2.1	0.12
AlbaR75 F2	2			15929	16550	1.1	16550	1.6	0.12
AlbaR77 F1	5			13916.5	16358*	0.9	16358*	0.8	
AlbaR77 F2	5			15639.5	16358*	0.7	16358*	0.9	

Table 5
MRPP instances from Madrigueras graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
MadrR33 F1	49	196	316	18947.5	20570*	2.2	20570*	2.8	
MadrR33 F2	49			19760	20570*	1.7	20570*	2.4	
MadrR35 F1	48			21315	23100*	0.9	23100*	1.4	
MadrR35 F2	48			22440	23100*	0.6	23100*	1.0	
MadrR37 F1	39			24937.5	26740*	0.8	26740*	0.9	
MadrR37 F2	39			26517.5	26740*	0.4	26740*	0.9	
MadrR53 F1	25	196	316	19803.7	21255*	1.3	21255*	1.4	
MadrR53 F2	25			20587.5	21255*	0.7	21255*	1.0	
MadrR55 F1	19			26460	32390*	2.2	32390*	2.4	
MadrR55 F2	19			32140	32390*	0.6	32390*	1.0	
MadrR57 F1	20			30035	34455*	1.5	34455*	1.7	
MadrR57 F2	20			34195	34455*	0.5	34455*	0.8	
MadrR73 F1	5	196	316	23410	26047.5	11	26047.5	8.8	0.12
MadrR73 F2	5			24947.5	26047.5	4.4	26047.5	5.7	0.12
MadrR75 F1	4			30650	39235*	4.9	39235*	4.8	
MadrR75 F2	4			38895	39235*	0.5	39235*	0.7	
MadrR77 F1	5			29875	32470*	1.2	32470*	1.2	
MadrR77 F2	5			31690	32470*	0.9	32470*	1.2	

Table 6
MRPP instances from Aldaya graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
AldaR33 F1	52	214	351	28782.8	31404*	4.6	31404*	7.4	
AldaR33 F2	52			30356.3	31404*	3.8	31404*	5.2	
AldaR35 F1	52			37264	41395*	1.0	41395*	1.7	
AldaR35 F2	52			40895	41395*	0.5	41395*	1.4	
AldaR37 F1	38			34410	37029	1.8	37029	2.7	0.03
AldaR37 F2	38			35949	37029	1.4	37029	2.7	0.03
AldaR53 F1	35	214	351	28506	30384.5	8.5	30392*	2.0	
AldaR53 F2	35			29501	30384.5	8.3	30392*	2.9	
AldaR55 F1	21			33849	36565*	2.4	36565*	2.3	
AldaR55 F2	21			35877	36565*	1.6	36565*	2.3	
AldaR57 F1	22			38425	42841*	1.4	42841*	1.5	
AldaR57 F2	22			42455	42841*	0.7	42841*	1.0	
AldaR73 F1	10	214	351	31610	34115*	3.9	34115*	3.0	
AldaR73 F2	10			33304	34115*	3.1	34115*	3.5	
AldaR75 F1	1			36610	40816*	6.1	40816*	6.0	
AldaR75 F2	1			39974	40816*	1.6	40816*	2.3	
AldaR77 F1	6			55006	64739*	3.7	64739*	3.7	
AldaR77 F2	6			64614	64739*	0.6	64739*	0.7	

Table 7
MGRP instances from Albaida graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
AlbaG33 F1	66	116	174	10120	11792.7	1.2	11796*	2.2	
AlbaG33 F2	66			10272	11792.7	0.9	11796*	2.4	
AlbaG35 F1	68			8890	11146	2.1	11166	2.9	0.16
AlbaG35 F2	68			9075	11146	1.5	11166	3.0	0.16
AlbaG37 F1	72			14613	17583	0.7	17651.6	2.2	0.06
AlbaG37 F2	72			14757	17583	0.7	17651	3.0	0.06
AlbaG53 F1	35	116	174	10107	11551*	0.6	11551*	0.6	
AlbaG53 F2	35			10595	11551*	0.6	11551*	1.0	
AlbaG55 F1	28			13357	14393	0.7	14445	0.7	0.14
AlbaG55 F2	28			13915	14393	0.5	14445	0.7	0.14
AlbaG57 F1	34			15475	17124*	1.3	17124*	1.3	
AlbaG57 F2	34			16406	17124*	0.9	17124*	1.5	
AlbaG73 F1	13	116	174	12550	14030*	1.0	14030*	1.0	
AlbaG73 F2	13			13142	14030*	0.5	14030*	0.7	
AlbaG75 F1	12			14235	16109*	1.3	16109*	1.3	
AlbaG75 F2	12			15271	16109*	0.7	16109*	0.8	
AlbaG77 F1	7			16052	18182*	1.3	18182*	1.2	
AlbaG77 F2	7			17458	18182*	0.9	18182*	1.3	

Table 8
MGRP instances from Madrigueras graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
MadrG33 F1	104	196	316	16740	17573.7	5.8	17631.7	112	0.83
MadrG33 F2	104			16915	17573.7	4.8	17623.2	79	0.88
MadrG35 F1	97			20425	21675	1.5	21735	5.3	0.30
MadrG35 F2	97			21020	21675	0.6	21735	6.0	0.30
MadrG37 F1	99			22555	23830	0.5	23853.3	2.5	0.03
MadrG37 F2	99			23110	23830	0.3	23853.3	2.6	0.03
MadrG53 F1	44	196	316	20235	21747.5	8.8	21785*	2.2	
MadrG53 F2	44			20822.5	21747.5	4.8	21785*	1.8	
MadrG55 F1	45			23437.5	26301.2	1.8	26338.7	3.5	0.06
MadrG55 F2	45			25695	26301.2	0.8	26342.5	3.0	0.05
MadrG57 F1	41			25070	27700*	2.1	27700*	2.9	
MadrG57 F2	41			27600	27700*	0.8	27700*	1.7	
MadrG73 F1	14	196	316	23297.5	26365*	3.9	26365*	4.0	
MadrG73 F2	14			25707.5	26365*	2.3	26365*	4.6	
MadrG75 F1	7			27315	33320	3.7	33355*	4.1	
MadrG75 F2	7			32697.5	33320	0.7	33355*	1.0	
MadrG77 F1	10			32420	35405*	2.4	35405*	2.4	
MadrG77 F2	10			35031.2	35405*	0.9	35405*	1.2	

Table 9
MGRP instances from Aldaya graph

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
AldaG33 F1	109	214	351	24178.5	24980.7	5.3	25050.5	29	0.64
AldaG33 F2	109			24231	24980.7	3.1	25050.5	21	0.64
AldaG35 F1	100			28842	31446	1.9	31652	11	0.66
AldaG35 F2	100			29982	31446	1.6	31652	13	0.66
AldaG37 F1	105			28315	31480	1.3	31538	9.8	0.01
AldaG37 F2	105			28919	31480	1.0	31538	9.7	0.01
AldaG53 F1	39	214	351	32146	36197	7.1	36213*	10	
AldaG53 F2	39			35286	36197	4.1	36209	9.0	0.01
AldaG55 F1	51			33055	36903*	2.4	36903*	2.5	
AldaG55 F2	51			36017.5	36903*	1.0	36903*	2.0	
AldaG57 F1	35			38366.5	40832*	1.4	40832*	1.9	
AldaG57 F2	35			40611	40832*	0.5	40832*	1.2	
AldaG73 F1	9	214	351	34020	36995*	4.1	36995*	3.5	
AldaG73 F2	9			35956	36995*	1.3	36995*	2.9	
AldaG75 F1	6			39761	47228*	6.4	47228*	6.2	
AldaG75 F2	6			46329	47228*	3.6	47228*	4.8	
AldaG77 F1	10			38655	41730*	1.5	41730*	1.6	
AldaG77 F2	10			41197	41730*	0.7	41730*	0.9	

All the tables contain two rows for each instance, presenting the results obtained from formulations F1 and F2. Columns labelled 'p', 'V_R' and '$E \cup A$' represent, respectively, the number of R-sets, the number of (required) vertices and the number of links of each instance. Column 'LB0' shows the initial lower bound (see Section 3.1). Columns 'LBpol' and 'Tpol' present the bound obtained with the cutting-plane algorithm using only the inequalities that are known to be separated in polynomial time (see Section 3.2) and the time used in seconds, respectively. Next columns 'LB' and 'Time' give, respectively, the lower bound and the corresponding time in seconds obtained with the cutting-plane algorithms described in Section 3.3. These two last columns are not shown for the MCPP instances because the additional valid inequalities used to obtain the final bound LB do not apply when all the links are required. An '*' means that an optimal solution (not only the optimal value) has been obtained. Finally, column 'Dev' shows the percentage deviation of the LB bound from the optimal value. Although we have not implemented a full Branch & Cut procedure, our cutting-plane algorithm can be used to obtain the optimal solutions. Note that when the procedure finishes before the optimal solution is reached, we can resort to the branch-and-bound option of CPLEX (that may generate general cutting planes). If the integer solution is a feasible solution, it is optimal. Otherwise, some inequalities violated by the integer solution can be detected and added to the last LP and the process continues. This is the procedure that, when needed, has been applied here to obtain the optimal solutions.

From the 27 MCPP instances reported in Tables 1–3 it is clear that the LB0 corresponding to F2 is much better than that of F1, as it should be the case, since it has been proved that $LP0_{F2}$ is stronger than $LP0_{F1}$. Moreover, the corresponding polynomial bounds (LBpol) are obviously equal, although the cutting-plane procedure based on F1 is in general more time consuming. In any case, both algorithms obtain the optimal solutions of these medium-size instances within a few seconds.

Table 10
Random MRPP instances

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
R422 F1	102	357	884	18005.5	18992.6	14.7	19009.7	45.1	0.03
R422 F2				18099	18992.6	15.8	19009.3	120	0.03
R425 F1				20828	22211.5	3.9	22234*	11.0	
R425 F2				21404	22211.5	4.3	22234*	12.7	
R427 F1				29328	30615	1.9	30615	8.2	0.11
R427 F2				30242.5	30615	2.2	30615	10.0	0.11
R452 F1	23	465	1055	27321	29359.5	726	29359.5	716	0.01
R452 F2				28264	29359.5	3042	29359.5	426	0.01
R455 F1				33112.2	35862.5	25.3	35862.5	28.8	0.09
R455 F2				34916.3	35862.5	24.9	35862.5	27.3	0.09
R457 F1				45038	49155	26.2	49162*	22.6	
R457 F2				48441	49155	12.5	49162*	13.5	
R472 F1	3	498	1114	37642.5	40744*	328	40744*	304	
R472 F2				39839	40744*	188	40744*	91.6	
R475 F1				43872.5	49972*	111	49972*	109	
R475 F2				49174.5	49972*	9.8	49972*	9.7	
R477 F1				59846.5	66481*	23.9	66481*	23.7	
R477 F2				66068	66481*	4.3	66481*	4.1	
R522 F1	79	388	1106	21249.7	22287.6	76.8	22292.5	135	0.12
R522 F2				21429	22287.6	273	22292.5	105	0.12
R525 F1				25537.3	26669*	26.1	26669*	92.4	
R525 F2				25965.8	26669*	75.8	26669*	64.1	
R527 F1				33867	36003.5	2.6	36038.4	46.0	0.09
R527 F2				35231.5	36003.5	3.9	36038.2	44.0	0.09
R552 F1	7	488	1318	34343.5	35958*	757	35958*	541	
R552 F2				34895.2	35958*	4106	35958*	672	
R555 F1				40408.6	42951	32.6	42951	33.4	0.01
R555 F2				42217.5	42951	245	42951	99.0	0.01
R557 F1				54120	57096*	8.0	57096*	8.4	
R557 F2				56745.5	57096*	5.8	57096*	6.0	
R572 F1	1	498	1326	48479	50304*	68.6	50304*	68.9	
R572 F2				49261.9	50304*	30.2	50304*	40.6	
R575 F1				53508	57880*	362	57880*	365	
R575 F2				56971.5	57880*	28.9	57880*	25.1	
R577 F1				65212.5	71769*	60.1	71769*	61.5	
R577 F2				71267.5	71769*	12.6	71769*	11.9	

The above comments are also valid for the MRPP instances in Tables 4–6. Note also that the polynomial lower bounds from F1 and F2 equal here the optimal values in 20 out of the 27 instances and that the final bound LB is better than LBpol since, within similar computing times, it obtains the previous 20 optimal values and 3 new ones. As expected, bounds LB from formulations F1 and F2 are similar. Finally,

Table 11
Random MGRP instances

	p	V_R	$E \cup A$	LB0	LBpol	Tpol	LB	Time	Dev (%)
G422 F1	245	500	1105	20536	21934.3	36.7	21990.8	2384	0.62
G422 F2				20552.7	21934.3	72.3	21987.5	4964	0.63
G425 F1				23520.5	25793.3	20.4	25827.6	731	0.41
G425 F2				24057	25793.3	26.4	25827.5	588	0.41
G427 F1				27252	29492	13.8	29513	699	0.31
G427 F2				27572	29492	20.4	29517.2	731	0.30
G452 F1	58	500	1118	29829	32667	36.6	32672*	30.4	
G452 F2				31362.5	32667	13.5	32672*	21.4	
G455 F1				33118.7	35963.5	189	35998.6	118	0.05
G455 F2				34951.1	35963.5	436	35998.6	183	0.05
G457 F1				40666.5	43025	6.6	43035*	11.0	
G457 F2				42352	43025	4.3	43035*	8.0	
G472 F1	5	500	1116	37471.5	39725	974	39725	251	0.00
G472 F2				38601.5	39725	1932	39725	788	0.00
G475 F1				42556.5	50307*	75.9	50307*	71.1	
G475 F2				49573.5	50307*	15	50307*	17.2	
G477 F1				52060.5	57952*	21.4	57952*	22.6	
G477 F2				57374	57952*	16.6	57952*	14.3	
G522 F1	191	500	1301	22762	24372.0	206	24490.3	3734	0.55
G522 F2				22963.5	24371.9	129	24478.5	6480	0.59
G525 F1				25729.5	26602.2	10.7	26651.2	183	0.32
G525 F2				25829.2	26602.2	11.5	26650.5	219	0.32
G527 F1				31177	32774.5	8.8	32802	129	0.08
G527 F2				31485	32774.5	7.3	32802	114	0.08
G552 F1	19	500	1339	34683	36394*	411	36394*	230	
G552 F2				35464.5	36394*	85.6	36394*	88.3	
G555 F1				41229.5	43243.5	19.3	43268	19.2	0.00
G555 F2				42573.8	43243.5	8.8	43268	7.9	0.00
G557 F1				54019	57414*	14.7	57414*	14.7	
G557 F2				57084	57414*	6.2	57414*	6.5	
G572 F1	3	500	1329	48360.5	50058*	1787	50058*	609	
G572 F2				48908.5	50058*	217	50058*	112	
G575 F1				53296	61128*	1384	61128*	1033	
G575 F2				60242	61128*	82.5	61128*	48.2	
G577 F1				71266	79352	79.6	79359*	87.6	
G577 F2				79123.5	79352	5.4	79359*	6.9	

with respect to the times needed to compute LBpol and LB, note that both algorithms require a similar computational effort.

With respect to the MGRP instances in Tables 7–9 note that, when bound LBpol does not reach an optimal solution, the final bound LB is better than bound LBpol and that the improvement is similar with both formulations.

In order to get a deeper understanding of the differences between the two formulations, we decided to generate larger MRPP and MGRP instances, as described above. Some of the generated instances are really big, having up to 245 *R*-sets, 500 vertices and 1339 links. We think that these are the biggest instances published so far.

To our surprise, the polynomial cutting-plane algorithms are able to optimally solve 9 out of the 18 MRPP instances and 6 out of the 18 MGRP instances, but these are those instances having the smaller number of *R*-sets. Note that when some of the inequalities separated through heuristics are included, as the final bound LB does, the optimal solution is also obtained in these instances with usually less computational effort. On the remaining instances bound LB outperforms the polynomial bound LBpol.

Finally, as mentioned at the end of Section 3.3, there are some differences between $\mathrm{LB}_{\mathrm{F1}}$ and $\mathrm{LB}_{\mathrm{F2}}$ but, as we expected, they are minimal. The small differences reported may be just due to differences in the sequence of the LP solutions found. Therefore, we can state that formulations F1 and F2 behave equal with respect to the final bounds obtained. As for the computing times, note that the LP's solved by the cutting-plane algorithm based on formulation F1 are smaller, since only one variable per edge is defined in F1, while the algorithm based on F2 does not require the identification of balanced-set inequalities at each iteration. Although on most of the instances the second algorithm terminates earlier, there are big differences between the effort taken by each algorithm on some of them. For example, in instances G422 and G522 in which the number of *R*-sets and required edges is big, the algorithm based on F1 is faster than that based on F2.

5. Conclusions

We summarize in this final section the conclusions of the comparison between the two proposed formulations for arc routing problems on mixed graphs. With respect to the initial bounds, it has been shown that $\mathrm{LP0}_{\mathrm{F2}}$ dominates the initial bound obtained from formulation F1. Furthermore, the former can be easily obtained and the computational results show that it is much better than bound $\mathrm{LP0}_{\mathrm{F1}}$. Then, if we are interested in a simple and good bound in order to test, for example, the quality of a feasible solution provided by a heuristic, we think that bound $\mathrm{LP0}_{\mathrm{F2}}$ should be our first choice.

Polynomial bounds LBpol obtained from formulations F1 and F2 have been shown to be equal, though the algorithm based on F2 is usually faster since it only needs to identify inequalities in two classes.

As for the inequalities included in the bound, it pays to look (heuristically) for *K-C*, $K\text{-}C_{02}$, Honeycomb and Honeycomb_{02} inequalities, since in both formulations the final bound LB improves on the polynomial bound, taking into account the values obtained and the computational times.

Bounds $\mathrm{LB}_{\mathrm{F1}}$ and $\mathrm{LB}_{\mathrm{F2}}$ obtained using the cutting-plane procedures from formulations F1 and F2 are generally equal. Therefore, it can be concluded that both formulations behave equal with respect to the final bounds. With respect to the algorithms providing them, we would like to point to two main differences. While, on one hand, the algorithm based on F1 deals with only one variable per edge and therefore the LP's to be solved are smaller, that based on F2 does not require the identification of balanced-set inequalities at each iteration and therefore it is easier to code and needs fewer cutting-plane iterations. However, although in some instances this latter algorithm terminates earlier, there is no a common trend with respect to the time needed to compute $\mathrm{LB}_{\mathrm{F1}}$ and $\mathrm{LB}_{\mathrm{F2}}$. Indeed, there are big differences between the effort taken by each algorithm on some instances.

Finally, as mentioned in Section 4.2, when a cutting-plane algorithm finishes before the optimal solution is reached, it is possible to invoke the branch-and-bound option of CPLEX in order to get an integer solution. In this case, an advantage of using formulation F2 is that when the integer solution is connected it is an optimal solution. This is not true if we use formulation F1, since in this case the integer solution should also be even and balanced.

Acknowledgements

Authors want to thank two anonymous referees for their careful reading of the manuscript. Also we thank the support given by the Ministerio de Ciencia y Tecnología of Spain through Grant TIC2003-05982-C05-01 and by the Generalitat Valenciana (Grupos 03/189).

References

[1] Dror M, editor. Arc routing: theory, solutions and applications. Dordrecht: Kluwer Academic Publishers; 2000.
[2] Papadimitriou CH. On the complexity of edge traversing. Journal of the Association for Computing Machines 1976;23: 544–54.
[3] Ford LR, Fulkerson DR. Flows in networks. Princeton, NJ: Princeton University Press; 1962.
[4] Christofides N, Benavent E, Campos V, Corberán A, Mota E. An optimal method for the mixed postman problem. In: Thoft-Christensen P, editor. System modelling and optimization. Lecture notes in control and information sciences, vol. 59. Berlin: Springer; 1984.
[5] Grötschel M, Win Z. A cutting-plane algorithm for the windy postman problem. Mathematical Programming 1992;55: 339–58.
[6] Nobert Y, Picard JC. An optimal algorithm for the mixed Chinese postman problem. Networks 1996;27:95–108.
[7] Corberán A, Romero A, Sanchis JM. The mixed general routing problem polyhedron. Mathematical Programming 2003;96:103–37.
[8] Corberán A, Mejía G, Sanchis JM. New results on the mixed general routing problem. Operations Research, 2005;53(5), to appear.
[9] Kappauf CH, Koehler GJ. The mixed postman problem. Discrete Applied Mathematics 1979;1:89–103.
[10] Ralphs TK. On the mixed Chinese postman problem. Operation Research Letters 1993;14:123–7.
[11] Christofides N, Campos V, Corberán A, Mota E. An algorithm for the rural postman problem. Report IC.OR. 81.5. Imperial College, London; 1981.
[12] Eiselt HA, Gendreau M, Laporte G. Arc-routing problems, Part 2: the rural postman problem. Operations Research 1995;43:399–414.
[13] Edmonds J, Johnson EL. Matching, Euler tours and the Chinese postman. Mathematical Programming 1973;5:88–124.
[14] Padberg MW, Rao MR. Odd minimum cut-sets and *b*-matchings. Mathematical Operations Research 1982;7:67–80.
[15] Grötschel M, Lovász L, Schrijver A. Geometric algorithms and combinatorial optimization. Berlin: Springer; 1993.
[16] Benavent E, Corberán A, Sanchis JM. Linear programming based methods for solving arc routing problems. In: Dror M, editor. Arc routing: theory, solutions and applications. Dordrecht: Kluwer Academic Publishers; 2000.
[17] Reinelt G. TSPLIB: a traveling salesman problem library. ORSA Journal of Computing 1991;3:376–84.