Lower Bounds and Heuristics for the Windy Rural Postman Problem

E. Benavent[1], A. Carrotta[2], A. Corberán[1]*, J.M. Sanchis[3] and D. Vigo[2]
[1] D.E.I.O., Universitat de València (Spain)
[2] D.E.I.S., University of Bologna (Italy)
[3] D.M.A., Universidad Politécnica de Valencia (Spain)

March 2003

Abstract

In this paper we present several heuristic algorithms and a cutting-plane algorithm for the Windy Rural Postman Problem. This problem contains a big number of important Arc Routing Problems as special cases and has very interesting real-life applications. Extensive computational experiments over different sets of instances are also presented.

Keywords: Rural Postman Problem, Windy Postman Problem, Windy Rural Postman Problem, Heuristics, Cutting Planes.

1 Introduction

In this work we study an interesting arc routing problem, the Windy Rural Postman Problem (WRPP). This problem can be briefly described as follows. Given an undirected graph $G = (V, E)$ with two nonnegative costs associated to each edge (corresponding to the costs of traversing edge $e = (i, j)$ from i to j and from j to i, respectively) and a subset E_R of "required" edges (representing those edges requiring some service to be done along them), the WRPP consists of finding a tour of minimum cost traversing each edge in E_R at least once. The study and resolution of the WRPP is of great interest both since it generalizes most of the known single-vehicle Arc Routing Problems and it is the mathematical model describing several real-life problems.

An interesting application of the WRPP is described in what follows. Some 3-dimensional structures, as bridges, buildings skeletons or metallic structures, need to be

*corresponding author: angel.corberan@uv.es

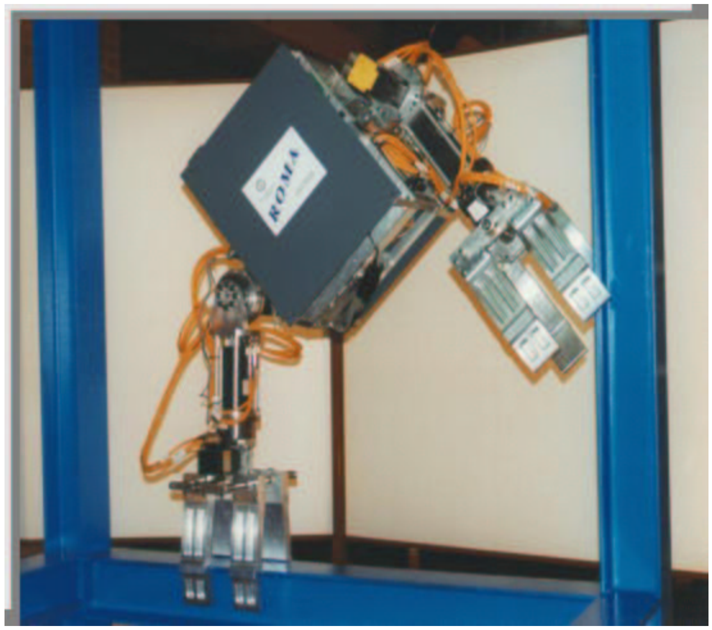

Figure 1: ROMA climbing robot

inspected periodically. The possibility of using autonomous robots, with TV cameras, for these tasks present a very important advantage with regard to safety and quality. During the last years several climbing robots have been developed. As an example, Figure 1 shows the ROMA robot designed in the University Carlos III of Madrid (see Balaguer et al., 2000). Due to the batteries they have incorporated, robots have a limited working time and therefore it is crucial to optimize their movements, i.e. the traversal of the structure should be of minimum length (Padrón, 2000). Figure 2 shows how these 3-dimensional structures can be modelled as undirected graphs. Dashed edges represent the 4 sides of each beam. Since all of them should be traversed for inspection, they will be considered required edges. On the other hand, solid and dotted lines in Figure 2 represent the different robot movements needed to change from one side of a beam to another side or to another beam. Since these edges can be traversed or not they will be considered as non required edges. Hence, the problem of finding a minimum energy consumption traversal passing through every required edge at least once can be modelled as a Rural Postman Problem. Moreover, since, for example, the energy consumed by the robot for moving down or up is different, the model should be extended to a Windy Rural Postman Problem.

If $E_R = E$, i.e. all the edges have to be traversed, the WRPP reduces to the Windy Postman Problem (WPP). The WPP was proposed in Minieka (1979) and, although in the general case is NP-hard (Brucker, 1981, Guan, 1984), it can be polynomially solved if the two possible orientations of each cycle have the same cost (Guan, 1984) or if G is an even (Eulerian) graph (Win, 1989). This last result is the basis for a heuristic algorithm proposed in Win (1989) to solve approximately the general case: first transform G into an even graph and then apply the exact algorithm for the WPP

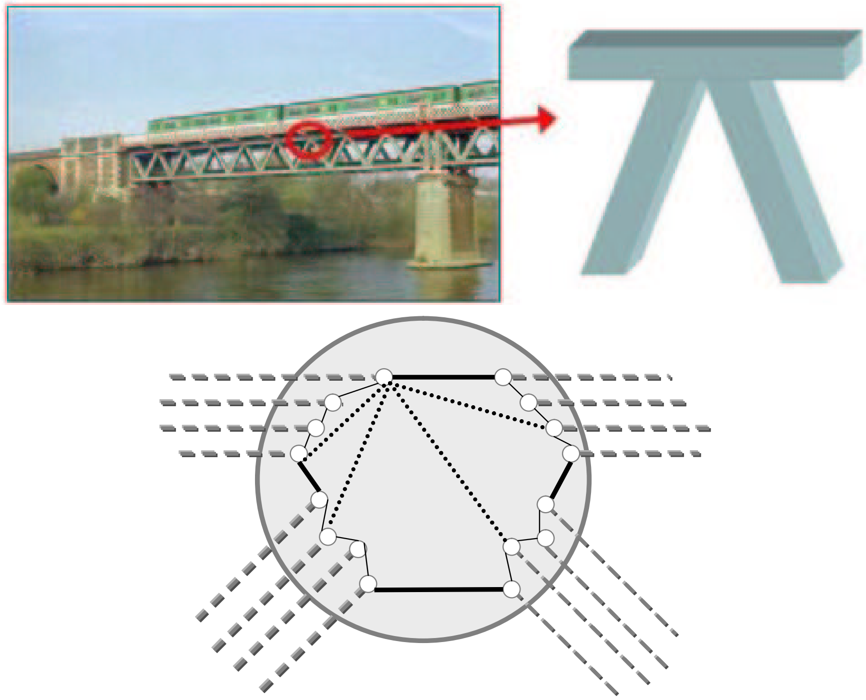

Figure 2: 3-D structures modelled as undirected graphs

on Eulerian graphs. In Pearn and Li (1994) another heuristic for the WPP is proposed that, basically, consists of executing the phases of the Win's algorithm in a different order. The chapter by Hertz and Mittaz (2000) in the recent book edited by Dror (2000) gives a nice description of these algorithms. An integer formulation of the WPP and a cutting-plane procedure for its resolution were proposed in Grötschel and Win (1992). In Laporte (1997) it is shown how several classes of Arc Routing Problems can be modelled and solved as directed Traveling Salesman Problems. This last approach seems to work well on graphs with few edges, which is not the case of the WPP and the WRPP.

By an appropriate definition of the costs c_{ij} and c_{ji} associated to each edge $e = (i, j)$, it is easy to see that the undirected, directed and mixed versions of the well known Chinese Postman Problem are particular cases of the WPP. Similarly, the Rural Postman Problem defined on an undirected, directed or mixed graph is a particular case of the WRPP. Therefore, the Windy Rural Postman Problem, is also NP-hard and, besides its practical applications, is the most general single-vehicle arc routing problem presented so far, and its study deserves considerable interest.

In Section 2 we extend the WPP formulation in Grötschel and Win (1992) to the Rural case and in Section 3 we present several families of valid inequalities that have been used in a cutting-plane procedure for the WRPP. Section 4 is devoted to describe three constructive heuristic algorithms. All the algorithms have been tested on a set of 288 WRPP instances described in Section 5. The computational results are presented in Section 6 and the conclusions in Section 7.

2 A formulation for the WRPP

In what follows we will assume, for the sake of simplicity, that all the vertices in V are incident with required edges. This is not a serious restriction as there is a simple way to transform WRPP instances which do not satisfy the assumption into instances which do (see, for instance, Christofides et al., 1986, or Eiselt et al., 1995). From the original graph $G = (V, E)$, we will define the graph G_R as the subgraph associated with the required edges, i.e., $G_R = (V, E_R)$. A WRPP solution is a strongly connected directed multigraph $G^* = (V, A)$ satisfying:

- each arc $(i, j) \in A$ is a copy of an edge in E with a given orientation,
- for each $(i, j) \in E_R$ either $(i, j) \in A$ or $(j, i) \in A$, and
- every vertex in G^* is symmetric (i.e., its indegree equals its outdegree).

It is well known that there exists a closed walk (tour) traversing each arc of G^* exactly once. This tour will be called in what follows a WRPP tour.

In general, graph G_R is not connected. The sets $V_1, V_2, \ldots, V_p$ of the connected components of G_R will be called R-sets. The subgraphs induced by these sets of vertices will be called R-connected components.

Given a node subset, $S \subseteq V$, let $\delta(S)$ denote the edge set with an end-point in S and the other in $V \setminus S$ and let $E(S)$ denote the edge set with both end-points in S. Given two node subsets $S, S' \subseteq V$, let $(S : S')$ denote to the edge set with one end-point in S and the other in S'. Finally, $\delta_R(S)$, $E_R(S)$, $(S : S')_R$ will denote to the previous sets referred only to the required edges.

Let x_{ij} be the number of times edge (i, j) is traversed from i to j in a WRPP tour. We propose the following formulation for the WRPP, where, for any subset $F \subseteq E$, $x(F)$ will denote $\sum_{(i,j)\in F} (x_{ij} + x_{ji})$:

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

s.t.:

$$x_{ij} + x_{ji} \geq 1, \qquad \forall (i,j) \in E_R \tag{1}$$

$$\sum_{(i,j)\in\delta(i)} (x_{ij} - x_{ji}) = 0, \qquad \forall i \in V \tag{2}$$

$$\sum_{i\in S, j\in V\setminus S} x_{ij} \geq 1, \qquad \forall S = \cup_{k\in Q} V_k, \quad Q \subset \{1,\ldots,p\} \tag{3}$$

$$x_{ij}, x_{ji} \geq 0 \tag{4}$$

$$x_{ij}, x_{ji} \ \text{ integer} \tag{5}$$

Inequalities (1) will be called traversing inequalities and imply that every required edge is traversed. Symmetry equations (2) force graph G^* to be symmetric, while connectivity inequalities (3) assure graph G^* to be connected. Note that, as constraints (1) imply the weak connectivity within the R-connected components, connectivity inequalities (3) can be restricted to the edge cut-sets between different R-sets. Note also that weak connectivity implies strong connectivity in a symmetric graph. This formulation is a generalization of that proposed by Grötschel and Win (1992) for the WPP.

3 Valid inequalities

We propose a cutting-plane procedure to exactly solve instances of the WRPP or, at least, to obtain lower bounds on the optimal cost. Let $\mathcal{X}$ be the set of all the WRPP tours, i.e., the set of all vectors $x \in I\!R^{2|E|}$ satisfying (1) to (5). Let us define the associated polyhedron WRPP(G)= $conv(\mathcal{X})$, i.e., the convex hull of the vectors in $\mathcal{X}$, whose (integral) extreme points correspond to the WRPP tours. We need some classes of valid inequalities for the polyhedron WRPP(G). To this end we will use families of valid inequalities that have been proved to be facet-inducing for the polyhedra associated to other routing problems related to the WRPP.

Given that $\mathcal{X}$ has been defined as the set of all vectors $x \in I\!R^{2|E|}$ satisfying (1) to (5), equations (2) are satisfied by all the points in WRPP(G) and inequalities (1), (3) and (4) are valid inequalities for WRPP(G).

Other families of valid inequalities can be obtained from the undirected RPP polyhedron. This polyhedron has been studied by Corberán and Sanchis (1994, 1998) and by Letchford (1997, 1999) and several families of facet-inducing inequalities have been described:

- the trivial inequalities,
- the connectivity inequalities,
- the R-odd cut inequalities,

- the K-C (K-Components) inequalities,
- the Honeycomb inequalities,
- the Path-Bridge inequalities.

Given an undirected graph $G = (V, E)$, a tour for the RPP is a vector $x \in \mathbb{Z}^{|E|}$ such that the graph containing x_e copies of each edge $e \in E$ is an even and connected graph containing all the required edges. Then, given a tour $x \in \mathbb{Z}^{2|E|}$ for the WRPP on G, the vector $x' \in \mathbb{Z}^{|E|}$, defined as $x'_e = x_{ij} + x_{ji}$ for all $e \in E$, is a tour for the (undirected) RPP on G. Hence, given a valid inequality for the RPP polyhedron

$$\sum_{e \in E} \alpha_e x'_e \geq \alpha_0,$$

the corresponding inequality

$$\sum_{e=(i,j) \in E} \alpha_e (x_{ij} + x_{ji}) \geq \alpha_0$$

is valid for the polyhedron WRPP(G).

Trivial and connectivity inequalities for the RPP correspond, respectively, to inequalities (4) and (3). The remaining inequalities above provide new families of valid inequalities for WRPP(G). In our cutting-plane algorithm we have only used the R-odd cut inequalities, the K-C inequalities and a special case of the Honeycomb inequalities.

The R-odd cut inequalities. Every WRPP tour traverses each edge cut-set of G an even number of times. If $\delta(S)$ is an edge cut-set with an odd number of required edges, then every WRPP tour must traverse at least the required edges plus an extra edge in $\delta(S)$. Then the following inequalities, called R-odd cut inequalities, are valid for WRPP(G):

$$x(\delta(S)) \geq |\delta_R(S)| + 1, \quad \forall S \subset V \text{ with } |\delta_R(S)| \text{ odd} \tag{6}$$

The K-C inequalities. They are defined in terms of an associated K-C configuration. A K-C configuration (see Figure 3) is a partition $\{M_0, \ldots M_K\}$ of V, with $K \geq 3$, such that

- $M_1, \ldots M_{K-1}$ and $M_0 \cup M_K$ are clusters of one or more R-sets,
- $|E_R(M_0 : M_K)|$ is positive and even,
- $E(M_i : M_{i+1}) \neq \emptyset$, for $i = 0, \ldots, K-1$.

The corresponding K-C inequality is:

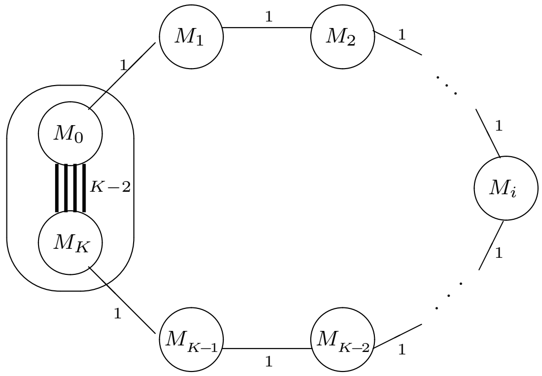

Figure 3: K-C configuration.

$$(K-2)\; x\Big((M_0\!:\!M_K)\Big) + \sum_{\substack{0 \le i < j \le K \\ (i,j) \ne (0,K)}} |\,i-j\,|\; x\Big((M_i\!:\!M_j)\Big) \ge \alpha_0 \qquad (7)$$

where the right hand side is $\alpha_0 = 2(K-1) + (K-2)|(M_0\!:\!M_K)_R|$.

The Honeycomb inequalities. They are a generalization of the previous K-C inequalities. In a K-C configuration, a R-connected component (or a cluster of R-connected components) is divided into two parts. In a *Honeycomb* configuration, this is generalized simultaneously both in the number L of R-connected components we divide and in the number of parts a R-connected component is divided. In general, Honeycomb configurations can be extremely complicated. Nevertheless, we restrict ourselves here to Honeycomb configurations where only one R-connected component is divided ($L\!=\!1$) into a number $\gamma \ge 2$ of parts. They consist of (see figure 4):

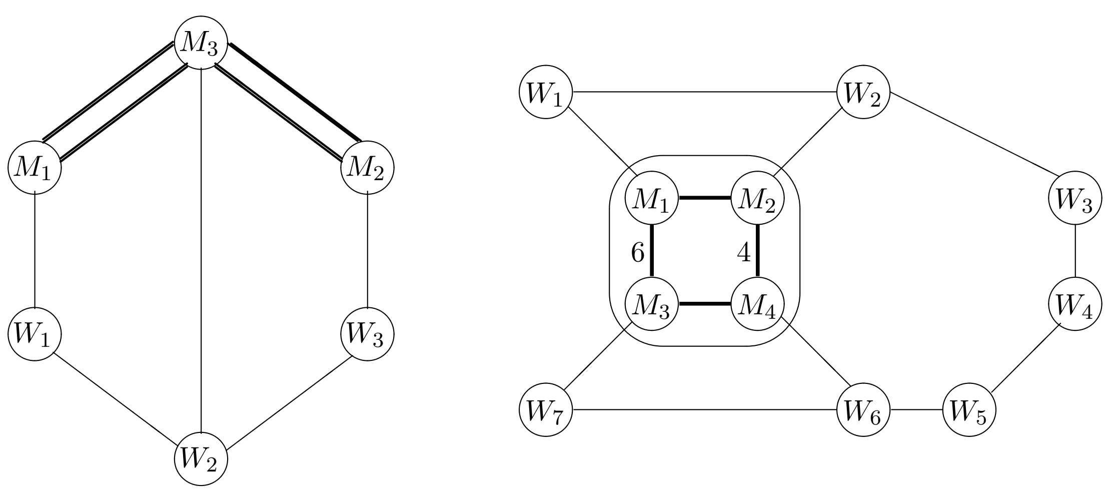

Figure 4: Two Honeycomb configurations.

- a partition $\{M_1, \ldots, M_\gamma, W_1, \ldots, W_{K-1}\}$ of V, with $\gamma \geq 2$, $K \geq 3$, such that $(M_1 \cup \ldots \cup M_\gamma)$, $W_1, \ldots, W_{K-1}$ are clusters of one or more R-sets, $\delta(M_i)$ contains a positive and even number of required edges for all i and the graph induced by the required edges on the vertex set $\{M_1, \ldots, M_\gamma\}$ is connected.
- a tree T spanning the sets $M_1, \ldots, M_\gamma, W_1, \ldots, W_{K-1}$ such that the degree in T of every vertex set M_i is 1, the degree of vertex sets W_j is at least 2 and the path in the tree connecting any distinct M_i, M_j is of length 3 or more.

For these Honeycomb configurations, the coefficient α_e of edge $e \in E$ in the associated Honeycomb inequality is equal to the number of edges traversed (if any) in the spanning tree to get from one end-vertex of e to the other, except for those edges with one end-vertex in M_i and the other in M_j, $i \neq j$, which coefficients are 2 units less. Then, all the coefficients associated to edges in the above figures are equal to 1, except where expressly shown. If we denote by α_R to the sum of these α-coefficients for all the required edges in the configuration, the Honeycomb inequality is then:

$$\sum_{e \in E} \alpha_e x_e \geq 2(K-1) + \alpha_R \tag{8}$$

Figure 4 shows two such Honeycomb configurations. The bold lines represent edges in $\delta_R(M_i)$ for some i and the thin lines represent edges in the spanning tree. The rhs of the associated inequality is 6+(1+1+1+1) for the configuration on the left and 14+(1+4+1+6) for the configuration on the right.

3.1 The cutting-plane algorithm

The cutting plane algorithm starts by solving an initial LP containing the objective function, all the inequalities (1), (2) and (4), a connectivity inequality (3) for each R-set V_i and a R-odd cut inequality (6) for each R-odd vertex of G. At each iteration, connectivity, R-odd cut, K-C and Honeycomb violated inequalities are found and added to the LP. Connectivity inequalities can be separated exactly in polynomial time by finding a minimum weight cut. The separation problem for inequalities (6) reduces to the problem of determining an odd-cut of minimum weight. Using a result of Padberg and Rao (1982), this problem can be solved exactly in polynomial time.

It is not known if the problem of separating K-C inequalities can be solved in polynomial time or not. However, Corberán, Letchford and Sanchis (2001) have designed a heuristic algorithm which works very well for the (undirected) RPP. They also separate the above Honeycomb inequalities with $L = 1$. These algorithms have been adapted here for the WRPP.

When the cutting-plane algorithm does not find more violated inequalities and the LP relaxation is still not integral, we invoke the branch-and-bound option of CPLEX. If the IP solution is a tour for the WRPP, we have obtained an optimal WRPP solution. Otherwise, the procedure terminates with a very tight lower bound.

4 Constructive Heuristics

In this section we will describe three heuristic algorithms. All the solutions generated by these algorithms are improved by means of three simple procedures that will be summarized at the end of the section. As in the preceding section, we will assume that all the nodes in graph $G = (V, E)$ are incident with required edges. The assumption is made only to simplify the notation since all the heuristics are implemented to deal with the case where it may exist nodes not incident with required edges.

We will denote by s_{ij} the cost of the shortest path in G from node i to node j. If a path traverses the edge set $P \subseteq E$, then, the *average cost* of that path, denoted by $\bar{s}_{ij}$, is defined as $\sum_{(i,j)\in P} (c_{ij} + c_{ji})/2$. Note that, since the shortest path from i to j will be, in general, different to that from j to i, $\bar{s}_{ij}$ and $\bar{s}_{ji}$ will also be different.

4.1 Heuristic H1

The first heuristic, called H1, extends to the WRPP the approaches proposed by Win (1989) for the WPP. In particular, the problem is solved by adding to graph G_R a set of edges that makes it connected. Then, a minimum cost matching problem is solved and further edges are added to obtain an even graph; afterwards, the solution of a minimum cost flow problem allows to construct a directed graph that is a solution of the WRPP. Then, the heuristic has several phases that are described in detail in what follows.

Phase 1: Shortest Spanning Tree

Recall that $V_1, V_2, \ldots, V_p \subseteq V$ denote the set of vertices of the connected components of the required subgraph G_R. If $p = 1$ go to Phase 2, otherwise,

i) build an undirected and complete graph H_1 whose nodes $w_1, w_2, \ldots, w_p$ represent the connected components of G_R and each edge (w_i, w_j) has a cost defined by: $\min\{\bar{s}_{pq}, \bar{s}_{qp} : p \in V_i, q \in V_j\}$;

ii) Compute a shortest spanning tree (SST) in H_1 and build the graph $G_1 = (V, E_R \cup E_1)$ where E_1 is determined as follows: for each (w_i, w_j) in the SST, add to E_1 a copy of each edge in the shortest path corresponding to edge (w_i, w_j). Therefore, the resulting G_1 is a connected graph. If G_1 has no odd-degree node, go to Phase 3; otherwise continue to Phase 2.

Phase 2: Minimum Cost Matching

Let us denote by $v_1, v_2, \ldots, v_k$ the odd degree nodes of G_1. Build a complete graph H_2 with this set of nodes and edges (v_i, v_j) with cost equal to $\min\{\bar{s}_{ij}, \bar{s}_{ji}\}$. Compute a minimum cost perfect matching in H_2, and build the set of edges E_2 by adding to it, for each edge in the optimal matching, a copy of all the edges in the corresponding shortest path. Therefore, the resulting $G_2 = (V, E_R \cup E_1 \cup E_2)$ is an even graph.

Phase 3: Win's algorithm for the WPP defined on an even graph

i) For each edge (i, j) of G_2, define its orientation from i to j if $c_{ij} \leq c_{ji}$, and from j to i otherwise, thus obtaining the arc set A_1. Define now an auxiliary network $H_3 = (V, A_1 \cup A_2 \cup A_3)$ with node set V, as follows. Each node $i \in V$ is assigned a demand b_i equal to the difference between its indegree minus its outdegree relative to arc set A_1. The arcs $(i, j) \in A_1$ have cost c_{ij} and capacity $+\infty$. Create two new sets of arcs A_2 and A_3, where for each arc $(i, j) \in A_1$:

   - include in A_2 an arc (j, i) with cost c_{ji} and capacity $+\infty$, and
   - include in A_3 and arc (j, i) with cost $(c_{ji} - c_{ij})/2$ and capacity 2.

ii) Solve a minimum cost flow problem on network H_3 with the above defined demands, capacities and costs. Let us denote the optimal flow through an arc (i, j) by f_{ij}, if $(i, j) \in A_1 \cup A_2$, and by f'_{ij}, if $(i, j) \in A_3$.

iii) Construct a new directed graph $G_3 = (V, A_4)$ as follows: for each arc $(i, j) \in A_3$,

   - if $f'_{ij} = 0$, put $f_{ji} + 1$ copies of arc (j, i) in A_4,
   - if $f'_{ij} = 2$, put $f_{ij} + 1$ copies of arc (i, j) in A_4.

   Since the demands are all even numbers, it can be shown that there always exists an optimal flow such that $f'_{ij} \in \{0, 2\}$, for all $(i, j) \in A_3$.

Graph G_3 is connected and symmetric and contains all the original required edges (as oriented arcs). Therefore, graph G_3 is a WRPP solution.

4.2 Heuristic H2

This heuristic consists of the same basic phases of H1 but applied in a different order. Thus, in the first two phases, a minimum cost flow problem and a matching problem are solved, thus obtaining a symmetric graph that may be not connected. These first two phases are similar to the algorithm Reverse-Win presented by Pearn and Li (1994) for the WPP. In the case where the resulting graph is not connected, a third phase is executed in which a shortest spanning tree is computed to connect it.

Phase 1 Minimum Cost Flow

i) For each edge in E_R, create an arc oriented according to its minimal traversing cost. Let $G_1 = (V, A_1)$ be the resulting directed graph. Let b_i the difference between the indegree and the outdegree of node i in G_1.

ii) Solve the Minimum Cost Flow Problem defined on the network $H_2 = (V, A_1 \cup A_2 \cup A_3)$, where:

   - the demand of each node $i \in V$ is b_i;
   - arcs $(i, j) \in A_1$ have cost c_{ij} and capacity $+\infty$;
   - A_2 contains an arc (j, i) with cost c_{ji} and capacity $+\infty$, for each arc $(i, j) \in A_1$;
   - A_3 contains an arc (j, i) with cost $(c_{ji} - c_{ij})/2$ and capacity 2, for each arc $(i, j) \in A_1$.

   Again, let f_{ij} be the flow through arc $(i, j) \in A_1 \cup A_2$ and let f'_{ij} be the flow through arc $(i, j) \in A_3$, in the optimal flow solution.

iii) Construct a mixed graph $G_3 = (V, E_1 \cup A_4)$, as follows: for each arc $(i, j) \in A_3$,

   - if $f'_{ij} = 0$, put $f_{ji} + 1$ copies of arc (j, i) in A_4,
   - if $f'_{ij} = 1$, put an edge (i, j) in E_1, and
   - if $f'_{ij} = 2$, put $f_{ij} + 1$, copies of arc (i, j) in A_4.

   Note that the directed graph induced in G_3 by the arc set A_4 is symmetric. If graph G_3 is even, then execute Phase 3, otherwise continue with Phase 2.

Phase 2 Minimum Cost Matching

i) Similarly to Phase 2 of heuristic H1, solve a Minimum Cost Matching Problem defined on a complete graph whose nodes are the odd degree nodes of G_3 and the edge costs are equal to the minimum of the average shortest path costs in G between its two end nodes. Then, for each edge in the optimal matching, add to E_1 a copy of each original edge in the corresponding shortest path. Let E_2 the resulting edge set; then, $G_4 = (V, E_2)$ is an undirected even graph.

ii) Apply Win's algorithm (Phase 3 of heuristic H1) to graph G_4 and let A_5 be the set of arcs in the solution. Then, $G_5 = (V, A_4 \cup A_5)$, is a symmetric graph that covers all the required edges at least once; if G_5 is connected, then it is a feasible WRPP solution, otherwise, it is necessary to execute Phase 3.

Phase 3 Shortest Spanning Tree

Solve a Shortest Spanning Tree defined on a complete graph whose nodes represent the connected components of G_5, and where edge costs are defined as the minimum of the average cost paths between nodes of different connected components (similarly to Phase 1 of heuristic H1). Then, consider a new arc set A_6, which is built as follows: for each edge of the solution tree, add to A_6 two opposite arcs for each original edge of the corresponding shortest path. Graph $G_6 = (V, A_4 \cup A_5 \cup A_6)$ is symmetric and connected and is a feasible WRPP solution.

4.3 Heuristic H3

This is a simple heuristic consisting of two phases. In the first one, a Shortest Spanning Tree, computed as in the first phase of heuristic H1, connects all the components of graph G_R. Then, for each edge e in the SST, add to G_R a copy of each edge in the shortest path corresponding to e. The resulting G_1 is an undirected and connected graph. At the beginning of second phase, edges in G_1 are oriented according to the minimum traversal cost, and demands b_i are computed as the difference between the arcs entering at and leaving from any node v_i in G_1. If $b_i > 0$, then v_i is a source node with supply b_i, while if $b_i < 0$, it is a sink node with demand $-b_i$. A Transportation Problem is defined with these supplies and demands, and with arc costs given by the shortest paths in G. Let f_{ij} be the solution to this problem, then add f_{ij} copies of arc (i, j) to G_1 to get a connected and symmetric directed graph that is a feasible WRPP solution.

4.4 Improvement procedures

Several simple improvement procedures are applied to the solutions obtained with the above three heuristics. The two first procedures look for cycles in the solution graph such that, either after deleting them or after reversing the direction of their arcs, a better solution is obtained. Let us denote by x_{ij} the number of times edge (i, j) is traversed from i to j in the present solution.

The first improvement strategy looks for simple cycles involving just two nodes and works as follows. Given two nodes i and j, such that $x_{ij} + x_{ji} \geq 3$, we compute

$$\delta = \begin{cases} min\{x_{ij}, x_{ji}\} & \text{if} \quad x_{ij} \neq x_{ji} \\ x_{ij} - 1 & \text{if} \quad x_{ij} = x_{ji} \end{cases}$$

It is obvious that by removing δ arcs from i to j and from j to i, we obtain another feasible solution with not greater cost (see Figure 5).

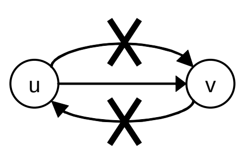

Figure 5: Improvement procedure 1

The second improvement procedure looks for more general cycles. The algorithm works on an auxiliary graph G' (see Figure 6) with the same nodes as the original graph and the following arcs:

a) For every arc (i,j) corresponding to a required edge, and such that $x_{ij} \geq 3$, put on G' an arc (j,i) with cost $-2c_{ij}$ and capacity $\lfloor \frac{x_{ij}-1}{2} \rfloor$.

b) For every arc (i,j) corresponding to a non-required edge, and such that $x_{ij} \geq 2$, put on G' an arc (j,i) with cost $-2c_{ij}$ and capacity $\lfloor \frac{x_{ij}}{2} \rfloor$.

c) For the remaining arcs (i,j) in the solution, put on G' an arc (j,i) with a cost of $c_{ji} - c_{ij}$ and capacity x_{ij}.

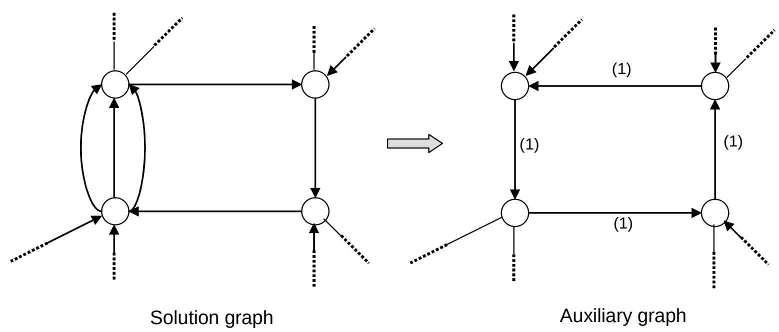

Figure 6: The auxiliary graph G'. Number in brackets represent arc capacities.

The first two types of arcs represent the possibility of removing a pair of arcs from the solution while the third type represent the cost of changing the orientation of the corresponding arc in the solution. Then solve a minimum cost flow problem on G', with demands and supplies equal to zero for all the vertices. Let f_{ij} be the flow associated to arc (i,j) in the optimal solution: we then perform the following changes to the WRPP solution graph (see Figure 7):

- if arc (i,j) has been included in G' due to conditions a) or b) above, then remove $2f_{ij}$ copies of arc (j,i) from the solution;
- if arc (i,j) has been included in G' due to condition c), then change the orientation of f_{ij} arcs (j,i) in the solution graph.

The solution graph obtained after the application of this second improvement procedure (see Figure 7) is symmetric and traverses all the required edges. Nevertheless, it can be non-connected. In such a case we compute a shortest spanning tree in a similar way to the constructive algorithms presented before. For each edge in the tree we add two opposite arcs to the solution graph to obtain a connected solution.

The third improvement procedure consists of first constructing an Eulerian tour traversing the solution graph and then by substituting any path containing only non required arcs and connecting two consecutive required edges by a shortest path.

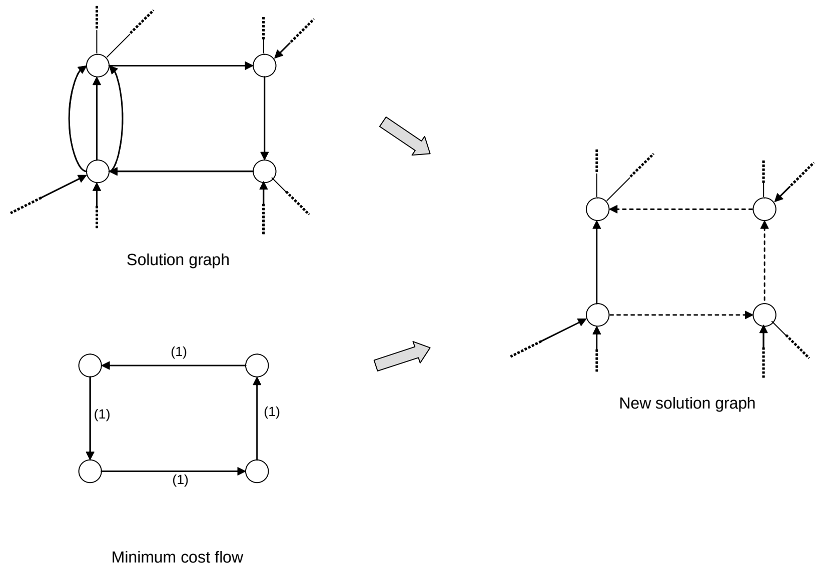

Figure 7: The new solution obtained after the second improvement procedure.

5 The test instances

We have tested our algorithms on three sets of randomly generated WRPP instances. Instances have been generated as follows.

The first set of instances were generated from the 24 RPP instances proposed in Christofides et al. (1981). These RPP instances have up to 84 vertices, 180 edges, 74 required edges and 8 R-connected components. Let us call C01 to C24 these RPP instances. From each of these 24 RPP instances we have generated 6 WRPP instances, by computing from each original cost c'_{ij}, two new costs c_{ij} and c_{ji} by means of the following strategies proposed in Win (1987):

Strategy 1. For each edge $e = (i, j) \in E$, let k_1 and k_2 be two integer values randomly selected in an interval $[-a, a]$, where $a = 5, 8$, and 10 as in Win (1987). Then, set $c_{ij} = \max\{1, c'_{ij} + k_1\}$, and $c_{ji} = \max\{1, c'_{ji} + k_2\}$.

Strategy 2. For each edge $e = (i, j) \in E$, let k_1 and k_2 be two integer values randomly selected in an interval $[a, b]$, where $[a, b] = [1, 100], [1, 200]$, and $[1, 500]$ as in Win (1987). Then, set $c_{ij} = k_1$, and $c_{ji} = k_2$.

In order to obtain larger WRPP instances, we have generated 144 more instances from two undirected graphs representing the real street networks of Albaida (Valencia, Spain) and Madrigueras (Albacete, Spain).

The graph of Albaida is an undirected graph with 116 vertices and 174 edges. In a first step, several RPP instances are generated from this graph as follows. Each original edge is selected as a required edge with probability p. In the case where some vertices were not incident with required edges, other edges are selected as required with the same probability and the procedure repeats until all the vertices are incident with required edges. Two RPP instances have been generated for each value of $p \in \{0.3, 0.5, 0.7\}$, named A31, A32, A51, A52, A71 and A72, where the first digit refers to the probability p, and the second one identifies the instance. All these instances have the original Albaida graph edge costs c_{ij}.

From each RPP instance two WRPP instances have been generated for each value of $a \in \{5, 8, 10\}$ (according to Strategy 1) and for each interval $[a, b] = [1, 100], [1, 200]$ and $[1, 500]$ (according to Strategy 2). Hence, we have 72 WRPP instances generated from the Albaida graph, all of them with 116 vertices and 174 edges.

The graph of Madrigueras is an undirected graph with 196 vertices and 316 edges. From this graph, we have generated 72 WRPP instances in the same way as for the Albaida instances.

6 Computational results

We now describe the results of the computational testing of the ANSI-C implementation of the cutting plane algorithm and of heuristics H1, H2, and H3 performed on a PC based on Pentium III 1GHz CPU.

Table 1 shows the computational results obtained on the WRPP instances generated from the Christofides et al. (1981) set of instances. Each row summarizes the results obtained for the 6 WRPP instances generated from each RPP instance. Column labelled '# of R-sets' shows the number of connected components of the graph induced by the required edges. It is well known that the difficulty of the Rural Postman Problem grows exponentially with the number of R-sets. The entries corresponding to heuristics H1, H2 and H3 are average percentage deviations from the lower bound obtained with the cutting plane procedure. For each heuristic, column (1) shows the results obtained without applying the improvement procedures described in Section 4.4, while column (2) shows the results obtained after applying them. Column (labelled 'Best') contains the average deviations obtained with a 'whole procedure' consisting of executing the three heuristics above and selecting the best solution obtained. The number of instances in which the cutting plane algorithm reached the optimal solution without invoking B&B is shown in column '# of LB opt.'. All the instances were solved to optimality with the B&B. Last column shows the averaged running time taken by the cutting plane (and B&B) procedure. Computing times for the heuristics are not shown as they are negligible (always less than 1 second). Last row shows the average results for all the instances.

Similarly, Tables 2 and 3 show the computational results obtained on the WRPP instances generated from the Albaida and Madrigueras graphs.

The results of the cutting plane are very good. It was able to solve up to optimality 185 out of 288 instances. For the unsolved instances, the average deviations from the

	# of inst.	# of R-sets	H1		H2		H3		Best	# of LB opt.	Time (scs.)
			(1)	(2)	(1)	(2)	(1)	(2)			
C01	6	4	0.75	0.75	13.69	8.47	22.16	6.04	0	6	2.2
C02	6	4	7.74	5.07	12.76	9.23	23.78	4.17	2.08	4	2.5
C03	6	4	9.63	6.53	8.14	5.71	29.92	7.59	3.59	3	3.4
C04	6	3	5.57	4.78	11.00	9.59	30.50	10.36	4.31	4	2.5
C05	6	5	7.43	5.40	19.95	10.92	42.18	5.67	2.07	4	3.1
C06	6	7	10.77	9.61	15.59	13.84	43.35	11.09	7.98	4	3.5
C07	6	3	5.75	3.14	6.41	0.89	32.20	9.05	0.89	6	2.0
C08	6	2	7.33	6.26	6.60	5.20	26.80	6.43	0.99	5	2.0
C09	6	3	8.40	3.27	10.88	3.96	25.11	2.43	1.35	3	2.7
C10	6	4	5.37	5.01	20.12	10.88	26.78	7.41	4.15	6	2.5
C11	6	3	3.11	3.11	12.73	7.52	46.72	8.68	1.28	5	1.7
C12	6	3	17.06	8.46	12.33	9.18	15.71	1.40	1.37	5	1.8
C13	6	3	5.31	5.31	14.20	8.75	29.17	10.28	2.93	6	1.4
C14	6	6	10.88	7.45	11.12	7.60	33.99	9.64	6.22	3	3.8
C15	6	8	2.80	1.73	11.42	8.11	17.09	6.11	1.73	6	3.1
C16	6	7	12.45	9.96	10.78	6.09	32.84	11.29	3.99	3	4.6
C17	6	5	12.63	10.75	17.44	11.34	28.49	11.45	6.65	5	2.1
C18	6	8	5.68	5.07	15.72	13.66	22.71	4.18	3.30	6	2.4
C19	6	7	8.94	7.60	16.69	15.45	31.29	6.51	5.02	4	5.5
C20	6	7	4.88	3.25	5.83	3.47	40.58	5.76	2.18	4	5.6
C21	6	6	5.93	3.31	8.43	6.64	36.18	10.72	3.23	3	6.5
C22	6	6	5.22	4.68	6.38	5.85	34.30	9.58	3.74	4	5.2
C23	6	6	5.34	3.88	5.66	4.28	24.26	4.72	2.61	4	6.6
C24	6	7	8.12	5.79	12.72	9.21	26.23	8.33	5.23	4	3.6
Global	144		7.37	5.42	11.93	8.15	30.12	7.45	3.20	107	3.3

Table 1: Computational results on the Christofides et al. (1981) instances

	# of inst.	# of R-sets	H1 (1)	H1 (2)	H2 (1)	H2 (2)	H3 (1)	H3 (2)	Best	# of LB opt.	Time (scs.)
A31	12	33	5.31	4.23	7.63	6.37	26.5	9.3	3.71	2	22.3
A32	12	28	5.02	4.15	6.31	4.48	28.9	11.1	3.34	8	17.2
A51	12	18	4.19	3.58	4.29	3.49	24.6	10.4	2.93	8	12.9
A52	12	21	5.04	3.71	5.24	4.14	28.8	9.3	3.15	8	16.6
A71	12	7	1.65	1.41	2.41	1.58	28.9	7.2	0.79	9	14.1
A72	12	8	1.92	1.66	1.93	1.67	25.9	8.9	0.84	10	17.4
Global	72		3.85	3.12	4.63	3.62	27.3	9.37	2.46	45	16.7

Table 2: Computational results on the Albaida instances

	# of inst.	# of R-sets	H1 (1)	H1 (2)	H2 (1)	H2 (2)	H3 (1)	H3 (2)	Best	# of LB opt.	Time (scs.)
M31	12	42	6.65	4.64	4.36	3.61	36.2	13.0	3.38	2	88.2
M32	12	47	8.93	7.28	3.80	2.89	35.1	14.7	2.89	1	109.2
M51	12	28	4.62	4.03	4.11	3.44	30.8	11.1	3.19	5	82.5
M52	12	22	3.89	3.42	3.02	2.37	31.9	11.0	2.23	6	42.2
M71	12	5	1.25	1.20	1.60	1.01	29.4	9.6	0.88	11	19.1
M72	12	8	2.18	2.01	2.37	1.64	29.8	9.5	1.45	8	31.0
Global	72		4.59	3.76	3.21	2.49	32.2	11.5	2.34	33	62.0

Table 3: Computational results on the Madrigueras instances

optimal value are 0.80% for the Christofides et al. instances, 0.37% for the Albaida instances and 0.18% for the Madrigueras instances. Furthermore, all these instances were solved when the B&B option of CPLEX was invoked.

On the other hand, as it can be seen in Tables 1 to 3, the improvement procedures, although simple, produce a significative decrease of the solution values. The solution average improvement is in the range 0.73% to 1.95% for heuristic H1, 0.72% to 3.78% for heuristic H2 and it reaches a 22.7% for heuristic H3. The simpler heuristic H3 produces, in general, the worst results. Heuristic H1 is the best in the Christofides et al. and in the Albaida instances, while heuristic H2 produces better results in the Madrigueras instances. Given that the computing times are very low, we think that the best strategy is to apply the three heuristics and choose the best solution obtained. In this case, as it can be seen in columns labelled 'best', we obtain average percentage deviations of 3.20, 2.46 and 2.34 for the three set of instances. We think these are satisfactory results for medium size instances.

7 Conclusions and future research directions

In this paper we deal with an interesting arc routing problem, the Windy Rural Postman Problem, that, besides its practical applications, generalizes most of the known arc routing problems with a single vehicle. Here, a formulation of the WRPP and three constructive heuristic algorithms and a cutting plane procedure have been proposed. Extensive computational results assessing the quality of the upper and lower bounds obtained have been reported.

Several directions of research about the WRPP are currently being developed. One consisting of the design of metaheuristics that based on the constructive algorithms here described can obtain high quality feasible solutions, although at a greater computational effort. A second one is devoted to the study of the polyhedron associated to the WRPP solutions. Finally, although the computational experiments reported above show that our cutting plane algorithm works well in practice, further improvements may still be achieved. Firstly, further separation routines for other known valid inequalities can be added. Secondly, one can embed the cutting plane procedure in a branch-and-cut scheme in such a way that optimality can always be guaranteed.

Acknowledgments: The contribution by E. Benavent, A. Corberán and J.M. Sanchis has been partially supported by the Ministerio de Ciencia y Tecnología of Spain (Ref: TIC2000-C06-01). A. Carrota and D. Vigo thanks the Ministero dell' Istruzione, dell'Università e della Ricerca (M.I.U.R.) and the Consiglio Nazionale delle Ricerche (C.N.R.), Italy, for the support given to this project.

References

[1] C. Balaguer, A. Giménez, J.M. Pastor, V.M. Padrón and M. Abderrahim, A climbing autonomous robot for inspection applications in 3D complex environments, *Robotica* 18, 287-297, 2000.

[2] P. Brucker, The Chinese Postman Problem for Mixed Graphs. In *Proc. Int. Workshop, Bad Honnef, 1980.* Lecture Notes in Computer Science, 100, 354-366, 1981.

[3] N. Christofides, V. Campos, A. Corberán and E. Mota, An algorithm for the Rural Postman Problem, *Report IC.O.R.81.5*, Imperial College, London, 1981.

[4] 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.

[5] A. Corberán, A. Letchford and J.M. Sanchis, A cutting plane algorithm for the General Routing Problem, *Mathematical Programming*, 90, 291-316, 2001.

[6] A. Corberán and J.M. Sanchis, A polyhedral approach to the Rural Postman Problem, *European Journal of Operational Research*, 79, 95-114, 1994.

[7] A. Corberán and J.M. Sanchis, The General Routing Problem Polyhedron: facets from the RPP and GTSP polyhedra, *European Journal of Operational Research*, 108, 538-550, 1998.

[8] M. Dror (Ed.), *Arc Routing: Theory, Solutions and Applications.* Kluwer Academic Publishers, 2000.

[9] H.A. Eiselt, M. Gendreau and G. Laporte, Arc-Routing Problems, Part 2: the Rural Postman Problem, *Operations Research*, 43, 399-414, 1995.

[10] M. Grötschel and Z. Win, A cutting plane algorithm for the Windy Postman Problem, *Mathematical Programming*, 55, 339-358, 1992.

[11] A. Hertz and M. Mittaz, Heuristic Algorithms. In *Arc Routing: Theory, Solutions and Applications* (M. Dror, ed.). Kluwer Academic Publishers, 2000.

[12] G. Laporte, Modeling and solving several classes of Arc Routing Problems as Traveling Salesman Problems, *Computers and Ops. Res.*, 24, 1057-1061, 1997.

[13] A.N. Letchford, New inequalities for the General Routing Problem, *European Journal of Operational Research*, 96, 317-322, 1997.

[14] A.N. Letchford, The General Routing Polyhedron: a unifying framework, *European Journal of Operational Research*, 112, 122-133, 1999.

[15] M. Guan, On the Windy Postman Problem, *Discrete Applied Mathematics*, 9, 41-46, 1984.

[16] E. Minieka, The Chinese Postman Problem for Mixed Networks, *Management Science*, 25, 643-648, 1979.

[17] V.M. Padrón, Herramientas de planificación de movimientos con restricciones para robots escaladores en estructuras tridimensionales complejas. *PhD Dissertation* (in spanish), Dept. de Ingeniería Eléctrica, Electrónica y Automática, University Carlos III of Madrid, Spain, 2000.

[18] W.L. Pearn and M.L. Li, Algorithms for the Windy Postman Problem, *Computers and Ops. Res.*, 21, 641-651, 1994.

[19] Z. Win, Contributions to routing problems. *PhD Dissertation*, University of Augsburg, Germany, 1987.

[20] Z. Win, On the Windy Postman Problem on Eulerian Graphs, *Mathematical Programming*, 44, 97-112, 1989.