Optimal Solutions for the Windy Rural Postman Problem by Branch & Cut

Angel Corberán* Isaac Plana* Jose M. Sanchis†

*Dept. d'Estadística i Investigació Operativa, Universitat de València
Av. Dr. Moliner, 50
46100 Burjassot(Valencia), Spain
{angel.corberan,isaac.plana}@uv.es

†Dept. de Matemática Aplicada, Universidad Politécnica de Valencia
Camino de Vera, s/n
46022 Valencia, Spain
jmsanchis@mat.upv.es

Abstract

In this paper we present an exact algorithm for the Windy Rural Postman Problem. This problem generalizes many important Arc Routing Problems, and also has some interesting real-life applications. The Branch & Cut method presented here has been tested over sets of instances of large size.

1 Introduction

The problem addressed in this paper, the Windy Rural Postman Problem (WRPP), can be described as follows. Let $G = (V, E)$ be an undirected graph with two costs c_{ij} and c_{ji} associated to each edge $(i, j) \in E$. The first one represents the cost of traversing this edge from vertex i to vertex j, while the second one represents the cost of the traversal in the opposite direction. Let E_R be a subset of edges, which will be called *required* edges. These ones will be the edges that must be obligatorily traversed by the solution. So the WRPP consists of finding a tour of minimum cost that traverses each required edge at least once.

The WRPP generalizes many well known Arc Routing Problems. If $E_R = E$, it reduces to the Windy Postman Problem (WPP), presented in [12] and studied in [15], [16], [10] and [14]. When $c_{ij} = c_{ji}$ for all edges, we have the Rural Postman Problem (RPP). If both conditions are present, the problem reduces to the Chinese Postman Problem. Also, versions of these problems on directed and mixed graphs can be found as particular cases of the WRPP.

The WRPP is also interesting because it is the optimization model describing some real-life situations. Let us take, for example, the case of some climbing robots designed to inspect complex 3-dimensional structures, such as bridges. They carry a limited battery, so their routes must be carefully designed in order to consume as less energy as possible. These robots are remotely controlled and are equipped with TV-cameras to inspect the structure in such a way any possible crack in the bridge beams, for example, can be detected. All beams must be inspected, so they can be represented by required edges. Some special movements must also be performed by the robots in order to move from the end of a beam to the beginning of another one or to another side of the same beam. These would be the non-required edges. Since, for example, the energy consumed by the robot is not the same if the movement is upwards or downwards, the cost of traversing each edge can be different for each direction. So the problem can be formulated as a WRPP.

Some heuristic algorithms and a cutting-plane procedure producing good lower bounds have been introduced in [1]. Also, metaheuristic approaches can be found in [2], particularly Multi-Start and Scatter Search algorithms. A detailed description of the WRPP polyhedron, as well as some new facet-defining inequalities are presented in [6] and [7]. To our knowledge, the algorithm presented here is the first exact procedure proposed for the WRPP.

In what follows, the notation that will be used along this paper and a linear integer formulation for the WRPP will be introduced, as well as some other families of facet inducing inequalities that will be used in our Branch & Cut (B&C) algorithm. Finally, some details of our B&C implementation and the computational results obtained on different sets of instances, including some of large size, will be presented.

2 Notation and WRPP formulation

For the sake of simplicity and without loss of generality, we will assume that every vertex in V is incident with at least one required edge. Given an undirected graph $G = (V, E)$ and a subset of required edges $E_R \subset E$, $\delta(i)$ and $\delta(S)$ will denote the sets of edges incident with vertex i and with vertices in S, respectively, while $\delta_R(i) = \delta(i) \cap E_R$ and $\delta_R(S) = \delta(S) \cap E_R$.

Let G_R be the graph induced by E_R. This graph is, in general, non connected, so we will call *R-sets*, and represent them by $V_1, V_2, \ldots V_p$, to the sets of vertices associated to the connected components of G_R. A set of vertices S for which $|\delta_R(S)|$ is odd will be called *R-odd set*, and $\delta(S)$ will be an *R-odd cutset*.

Let us represent by x_{ij} (x_{ji}) the number of times that a solution x traverses edge (i, j) from i to j (from j to i). Then, the WRPP can be formulated ([1]) as follows:

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

$$s.t.: \quad x_{ij} + x_{ji} \geq 1 \quad \forall (i,j) \in E_R \tag{2}$$

$$\sum_{(i,j)\in\delta(i)} (x_{ij} - x_{ji}) = 0 \quad \forall i \in V \tag{3}$$

$$\sum_{i\in S, j\in V\setminus S} x_{ij} \geq 1, \quad \forall S = \cup_{k\in Q} V_k, \quad Q \subset \{1, \ldots, p\} \tag{4}$$

$$x_{ij}, x_{ji} \geq 0 \qquad \forall (i,j) \in E \tag{5}$$
$$x_{ij}, x_{ji} \text{ integer} \qquad \forall (i,j) \in E \tag{6}$$

Inequalities (2) assure that each required edge is traversed at least once by the solution. Flow conservation equations (3) state that the number of times that a solution enters a vertex must be equal to the number of times it goes out from it. Finally, inequalities (4) are the *connectivity* constraints, also called *subtour elimination* constraints.

It can be shown that, under mild conditions, inequalities (2) and (5) induce facets of the WRPP polyhedron. The following families of inequalities also induce facets of the WRPP polyhedron (see [6]).

- **R-odd cut** inequalities: these inequalities were introduced in [8] for the RPP.
- **K-Component** or **K-C** inequalities: also presented in [8]. A variation on these inequalities, called K-C_{02} inequalities, are also used.
- **Honeycomb** or **HC** inequalities: they are a generalization of K-C inequalities, and were first presented in [9] for the General Routing Problem (GRP). HC_{02} inequalities are another variation on the HC inequalities that are used.
- **Path-Bridge** or **PB** inequalities: a different generalization of K-C inequalities. They were introduced in [11] for the GRP.
- **Zigzag** or **Z** inequalities: this is a new family of facet-inducing inequalities, and its detailed description can be found in [7].

3 Branch & Cut algorithm

In this section we present some details on our implementation of the B&C method. B&C algorithms were first introduced by Padberg and Rinaldi [13], and have shown to be among the most efficient methods to solve NP-hard problems to optimality. Basically, a B&C algorithm consists of a cutting-plane procedure working at the nodes of a Branch & Bound tree. At each node, facet-defining inequalities that are violated by the current LP solution are identified by the *separation algorithms*.

In our B&C algorithm, the initial LP at the root node is defined by constraints (2), (3), (5), one connectivity constraint (4) for each R-set and one R-odd cut inequality for each R-odd vertex. Note that, even in the case that an integer solution is found, it may be unfeasible. An exact method must then be applied in order to find if there is any connectivity constraint violated by this solution.

In the cutting-plane procedure, we have used exact separation algorithms for connectivity and R-odd cut constraints identification. We have also used heuristics for separating them, as well as for separating all other types of inequalities presented before. Except for the separation of Zigzag inequalities, which algorithm was designed specifically for this problem, the other separation problems were solved by means of algorithms that were adapted from previously existing algorithms for other Arc Routing Problems ([4], [5]).

In each iteration of the cutting-plane procedure, the separation algorithms are invoked as follows:

1. Connectivity and R-odd cut heuristics.
2. Exact connectivity separation if the heuristics failed.
3. Exact R-odd cut separation if the heuristics failed.
4. If the number of violated constraints found so far is at least 10, stop.
5. Zigzag, K-C and K-C_{02} heuristics.
6. If the number of violated constraints found so far is at least 10, stop.
7. HC and HC_{02} heuristics.
8. If the number of violated constraints found so far is at least 15, stop.
9. PB heuristics.

The procedure applies until no new violated inequalities are found or a *tailing-off* criterium is met.

In our B&C algorithm we use inequalities to perform branching. Let x^* be the Lp solution at a given node of the branching tree. Let S be the set of vertices of a R-set such that $2k < x^*(\delta(S)) < 2k + 2$. Since $x(\delta(S))$ should be an even number for every WRPP tour x, we can split the tours into those that satisfy $x^*(\delta(S)) \leq 2k$ and those satisfying $x^*(\delta(S)) \geq 2k+2$. This is done by adding the appropriate inequality to the current Lp. This is the strategy used in [3]. When no such a subset S is found, the branching is done on the "most non integer" variable.

For very large instances, the number of inequalities that are found during the exploration of the B&C tree is so big that, in some cases, the computer may not be able to find the optimal solution due to memory limitations, and very often computing time increases because of this problem. It is then important to find a way to store constraints that uses as less memory as possible. Using flow conservation equations (3), connectivity, R-odd cut, K-C and K-C_{02} inequalities can be transformed so that the number of non zero coefficients reduces to half of them in their original form. Furthermore, during its execution, our algorithm adds to the formulation a large number of R-odd cut inequalities, many of which are very similar to each other. While they use a lot of memory, the presence of inequalities that are so similar does not seem to prove of much value to the effectiveness of the algorithm. For this reason only a small number of the R-odd cut constraints that are identified by the exact separation algorithm are added.

In order to avoid that the separation algorithms spend too much time at each node of the B&C tree, we have implemented a *tailing-off* strategy. After each 5 consecutive iterations of all the separation algorithms at the same node of the tree, if the total increase of the value of the objective function during these 5 iterations is less than 0.01%, a branching step is done and the actual node is no longer explored.

	Time	Nodes	R-odd	Conn.	K-C	HC	PB	Z	Total
Albaida	0.62	4.86	81.51	17.33	26.54	6.87	1.03	1.93	135.22
Madrigueras	8.71	21.29	280.03	17.51	36.20	6.45	1.31	3.94	345.43
A_{500}	68.72	49.19	628.19	24.81	13.00	0.71	1.07	2.59	670.37

Table 1: Computational results

4 Computational results

The present B&C algorithm has been implemented in C/C++ using CPLEX 9.0 Concert Technology, and compiled with MS Visual C++.NET. The following tests were run on a Pentium IV at 1.7Ghz, with 512Mb RAM. The algorithm has been tested on a total set of 120 randomly generated instances with the following characteristics:

- A set *Albaida* of 72 instances with 116 vertices, 174 edges and from 7 to 33 R-sets. These instances are based on the real street network of Albaida (Valencia). Details on the generation of these instances can be found on [1].
- A set *Madrigueras* with 72 instances with 196 vertices, 316 edges and from 5 to 47 R-sets. They are also based on a real street network (see [1]).
- A set A_{500} of 24 instances with 265 to 488 vertices, from 842 to 1719 edges and between 1 and 76 R-sets. A detailed description on how they were generated can be found in [2].

Table 1 contains the results obtained on these sets of instances. It shows the average time (in seconds) needed to optimally solve the instances of each set, the average number of nodes explored in the B&C tree, and the average number of cuts of each type identified by the separation algorithms. Column labelled K-C (HC) shows the number of K-C and K-C_{02} (HC and HC_{02}) constraints found, and the last column presents the total number of violated inequalities identified.

Acknowledgments: Authors thank the support given by the Ministerio de Ciencia y Tecnología of Spain (Ref: TIC2000-C06-01) and by the AVCiT de la Generalitat Valenciana (Ref: GRUPOS03/189).

References

[1] E. Benavent, A. Carrotta, A. Corberán, J.M. Sanchis and D. Vigo, "Lower Bounds and Heuristics for the Windy Rural Postman Problem", Working Paper TR03-2003, Department of Statistics and Operations Research. University of Valencia, Valencia, Spain, 2003.

[2] E. Benavent, A. Corberán, E. Piana, I. Plana and J.M. Sanchis, "New Heuristic Algorithms for the Windy Rural Postman Problem", To appear in Computers and O.R., 2003.

[3] J.M. Clochard and D. Naddef, "Using path inequalities in a branch-and-cut code for the symmetric traveling salesman problem", In L. Wolsey and G. Rinaldi, editors, Proc. 3rd IPCO Conference, 291-311, 1993.

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

[5] A. Corberán, G. Mejía and J.M. Sanchis, "New Results on the Mixed General Routing Problem", Working Paper TR05-2002, Department of Statistics and OR, University of Valencia, Valencia, Spain, 2002.

[6] A. Corberán, I. Plana and J.M. Sanchis, "On the Windy Rural Postman Polyhedron", Working Paper, Department of Statistics and Operations Research. University of Valencia, Valencia, Spain, 2004.

[7] A. Corberán, I. Plana and J.M. Sanchis, "Zigzag inequalities: a new class of facet-inducing inequalities for Arc Routing Problems". Working Paper, Department of Statistics and Operations Research. University of Valencia, Valencia, Spain, 2004.

[8] A. Corberán and J.M. Sanchis, "A Polyhedral Approach to the Rural Postman Problem", *European J. of Operational Research* 79, 95-114 (1994).

[9] A. Corberán and J.M. Sanchis, "The General Routing Problem Polyhedron: Facets from the RPP and GTSP polyhedra", *European J. of Operational Research* 108, 538-550 (1994).

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

[11] A.N. Letchford, "New Inequalities for the General Routing Problem", *European J. of Operational Research* 96, 317-322 (1997).

[12] E. Minieka "The Chinese Postman for Mixed Networks", *Management Science* 25, 643-648 (1979).

[13] M.W. Padberg, G. Rinaldi, "Optimization of a 532-city symmetric traveling salesman problem by branch-and-cut", *Operations Research Letters* 6, 1-7 (1987).

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

[15] Z. Win, "Contributions to Routing Problens", Ph.D. thesis, University of Augsburg, Germany (1987).

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