New results on the Windy Postman Problem

Angel Corberán[1]*, Marcus Oswald[2], Isaac Plana[3], Gerhard Reinelt[2], José M. Sanchis[4]

[1] Dept. d'Estadística i Investigació Operativa, Universitat de València, Spain

[2] Inst. of Computer Science, University of Heidelberg, Germany

[3] Dept. de Matemàtiques per a l'Economia i l'Empresa, Universitat de València, Spain

[4] Dept. de Matemática Aplicada, Universidad Politécnica de Valencia, Spain

Abstract

In this paper, we study the Windy Postman Problem. This is a well-known Arc Routing Problem that contains the Mixed Chinese Postman Problem (MCPP) as a special case. We extend to arbitrary dimension some new inequalities that complete the description of the polyhedron associated with the Windy Postman Problem (WPP) over graphs with up to four vertices and ten edges. We introduce two new families of facet-inducing inequalities and prove that these inequalities, along with the already known odd zigzag inequalities, are mod-4 inequalities. Moreover, a branch-and-cut algorithm that incorporates two new separation algorithms for all the previously mentioned inequalities and a new heuristic procedure to obtain upper bounds are presented. Finally, the performance of a branch-and-cut algorithm over several sets of large WPP and MCPP instances, with up to 3000 nodes and 9000 edges (and arcs in the MCPP case), shows that, to our knowledge, this is the best algorithm to date for the exact resolution of the WPP and the MCPP.

Key Words: Polyhedral Combinatorics, Facets, Arc Routing, Windy Postman Problem, Mixed Chinese Postman Problem.

1 Introduction

In this paper, we discuss the Windy Postman Problem (WPP). This problem can be defined as follows. Given an undirected and connected graph $G = (V, E)$ with two non-negative costs c_{ij} and c_{ji} associated with each edge $\{i, j\} \in E$ corresponding to the cost of traversing it from i to j and from j to i, respectively, the WPP is to find a minimum cost tour on G traversing each edge at least once. This problem was introduced by Minieka ([14]). It is NP-hard in general ([2, 13]) and can be solved in polynomial time if G is Eulerian ([20]), if the cost of the two opposite orientations of every cycle in G is the same ([13]) or if G is a series-parallel graph ([22]).

In [12, 20], a cutting-plane algorithm for the WPP based on a previous polyhedral study was proposed that, as far as we know, was the first polyhedral approach that had been applied to the resolution of an NP-hard Arc Routing Problem. The authors proved that the odd-cut inequalities and the k-wheel inequalities are facet-inducing, although only the separation of the first ones was implemented in their cutting-plane algorithm. This algorithm was tested on 36 WPP instances with $52 \leq |V| \leq 264$ and $78 \leq |E| \leq 489$ and it provided an optimal solution for

*corresponding author: angel.corberan@uv.es

31 instances. More recently, a new family of facet-inducing inequalities for the Windy General Routing Problem, the odd zigzag inequalities, which also applies to the WPP, has been presented in [7]. These inequalities generalize the 3-wheel inequalities proposed in [20] for the WPP.

The WPP contains the mixed version of the well-known Chinese Postman Problem (MCPP) as a special case arising when, for each edge $\{i,j\} \in E$, either $c_{ij} = c_{ji}$ or $\max\{c_{ij}, c_{ji}\} = \infty$. As with the WPP, this NP-hard problem ([17]) can be handled with a formulation using two variables associated with each edge ([6, 18]) and therefore can be considered a WPP in which some variables have infinite cost. Hence, the results presented here for the WPP can be directly applied to the MCPP.

In the next section, we define the problem, introduce the notation that will be used in this paper and present some known results. In Section 3, we present the full description of the polyhedra associated with the WPP over graphs with up to four vertices and ten edges and we generalize the new inequalities found, obtaining two new families of facet-inducing inequalities, the even-even and odd-odd zigzag inequalities. We prove that these inequalities and the odd zigzag inequalities are mod-k inequalities in Section 4. The branch-and-cut algorithm for the resolution of the WPP is described in Section 5. Finally, the computational experiments performed on a set of large WPP and MCPP instances are described in Section 6.

2 Problem formulation and known results

Let $G = (V, E)$ be the graph for which a minimum cost WPP tour has to be determined. For $S \subseteq V$, let $\delta(S)$ denote the set of edges with one end-point in S and the other in $V \setminus S$ and let $E(S)$ be the set of edges with both end-points in S. For $S_1, S_2 \subseteq V$, (S_1, S_2) denotes the set of edges with one end-point in S_1 and the other in S_2. A vertex is called *even (odd)* if it is incident with an even (odd) number of edges. A subset $S \subset V$ is called *even (odd)* if it contains an even (odd) number of odd vertices.

Let x_{ij} be the number of times edge $\{i,j\}$ is traversed from i to j in a WPP tour. For $F \subseteq E$, we define $x(F) = \sum_{\{i,j\}\in F}(x_{ij} + x_{ji})$, and for (S_1, S_2), we define

$$x(S_1 : S_2) \;=\; \sum_{i\in S_1,\, j\in S_2} x_{ij}.$$

Note that $x(S_1, S_2) = x(S_1 : S_2) + x(S_2 : S_1)$.

The IP formulation of the WPP in [12, 20] is:

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

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

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

$$x_{ij}, x_{ji} \geq 0 \qquad \forall \{i,j\} \in E, \tag{3}$$

$$x_{ij}, x_{ji} \ \text{ integer} \qquad \forall \{i,j\} \in E, \tag{4}$$

where conditions (1) imply that each edge will be traversed at least once and conditions (2) force the (directed) graph represented by the tour to be symmetric, i.e., a graph in which the indegree of every vertex is equal to its outdegree. The above system includes an equation associated with each vertex. The $|V|$ equations (2) will be referred to as the *symmetry equations* and any $|V|-1$ of them are linearly independent.

Let $\mathrm{WPP}(G) \subseteq \mathbb{Z}^{2|E|}$ be the convex hull of vectors satisfying (1) to (4). In [11], it is shown that $\mathrm{WPP}(G)$ is an unbounded polyhedron, with dimension $2|E|-|V|+1$, and that the following inequalities are, under mild conditions, facet-inducing:

– trivial inequalities (3),

– traversing inequalities (1),

– odd-cut inequalities

$$x(\delta(S)) \geq |\delta(S)| + 1, \quad \forall\ S \subset V, \quad |\delta(S)| \ \text{ odd}, \tag{5}$$

– k-wheel inequalities.

Because of the symmetry equations (2) the odd-cut inequalities can be written with fewer non-zero elements as:

$$x(S : V \setminus S) \geq \frac{|\delta(S)|+1}{2}, \quad \forall\ S \subset V,\ |\delta(S)| \text{ odd}. \tag{6}$$

A new family of facet-inducing inequalities for the Windy General Routing Problem, the odd zigzag inequalities, which also applies to the WPP, has been presented in [7]. These inequalities generalize the 3-wheel inequalities.

In [10], it is shown that all the facet-inducing inequalities for $\mathrm{WPP}(G)$, with the exception of (3) and (1), are *weak configuration inequalities.* A valid inequality $\alpha x \geq \beta$ for $\mathrm{WPP}(G)$ is called a weak configuration inequality if there is a partition $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_r\}$ of V such that the subgraphs $G(\mathcal{B}_k)$ are connected and the variables associated with edges in the sets $E(\mathcal{B}_k)$ have coefficient zero in the inequality. This definition is based on that of *configuration inequalities* by Naddef & Rinaldi ([15]). They only differ in the fact that, in a configuration inequality, all the variables x_{uv}, associated with the edges $\{u, v\}$ with $u \in B_i$ and $v \in B_j$, have equal coefficients in the inequality, while a weak configuration inequality can have variables x_{uv} and x_{st}, with $u, s \in B_i$ and $v, t \in B_j$, with different coefficients. This is the case with the odd zigzag inequalities and the inequalities presented later in this paper. Naddef & Rinaldi ([15]) also introduced the notion of the *configuration graph* $G_{\mathcal{C}} = (V_{\mathcal{C}}, E_{\mathcal{C}})$. This is the graph resulting from shrinking vertex sets $\mathcal{B}_k$, $k = 1, \ldots, r$, into a single vertex each. In [10], a "lifting" theorem is proven stating that if a weak configuration inequality is facet-inducing for $\mathrm{WPP}(G_{\mathcal{C}})$, then it is also facet-inducing for $\mathrm{WPP}(G)$.

Because we present in this paper a separation algorithm for the odd zigzag inequalities, we describe them briefly in the context of the WPP. This class of valid inequalities is violated by fractional solutions containing a "zigzag" associated with variables with value 0.5 such as the one shown in Figure 1a. Consider a partition of the set of vertices V into 4 parts, M^1, M^2, M^3, and M^4, where each M^i contains an odd number of odd vertices. We define $\mathcal{H} = (M^1, M^2) \cup (M^3, M^4)$ (horizontal edges) and $\mathcal{D} = (M^2, M^3) \cup (M^1, M^4)$ (diagonal edges). Note that $\mathcal{H} \cup \mathcal{D} = \delta(M^1 \cup M^3)$. Let us assume we have a subset of edges $\mathcal{F} \subset (\mathcal{H} \cup \mathcal{D})$ satisfying $|\mathcal{H} \setminus \mathcal{F}| + |\mathcal{D} \cap \mathcal{F}| = |\mathcal{D} \setminus \mathcal{F}| + |\mathcal{H} \cap \mathcal{F}|$, or, equivalently,

$$|\mathcal{H}| + |\mathcal{D}| = 2|\mathcal{H} \cap \mathcal{F}| + 2|\mathcal{D} \setminus \mathcal{F}| \tag{7}$$

(see Figure 1b, where edges in $\mathcal{F}$ are represented in bold lines).

The configuration graph $G_{\mathcal{C}}$ associated with the odd zigzag inequalities is defined by the partition of V and the set $\mathcal{F}$ above, and by the following pair of coefficients associated with each edge e_{ij} (see Figure 1b):

$$(\alpha_{ij}, \alpha_{ji}) = \begin{cases} (0,2), & \forall e_{ij} \in \mathcal{H} \setminus \mathcal{F}, i \in M^1 \cup M^3, j \in M^2 \cup M^4, \\ (2,2), & \forall e_{ij} \in \mathcal{H} \cap \mathcal{F}, \\ (1,3), & \forall e_{ij} \in \mathcal{D} \cap \mathcal{F}, i \in M^1 \cup M^3, \quad j \in M^2 \cup M^4, \\ (1,1), & \text{otherwise.} \end{cases}$$

In [7], the following proposition is proved:

Proposition 1 *For M^1, M^2, M^3, and M^4 as above the* odd zigzag inequality *requires*

$$\begin{aligned} & x(\delta(M^1 \cup M^2)) + 2x(M^2 : M^1) + 2x(M^4 : M^3) + 2x(F_{zz}) \\ \geq \ & |(M^1, M^3)| + |(M^2, M^4)| + |(M^1, M^4)| + |(M^2, M^3)| + 2|\mathcal{H} \cap \mathcal{F}| + 2, \end{aligned} \tag{8}$$

for each $x \in WPP(G)$, where $x(F_{zz})$ denotes the variables associated with the edges in $\mathcal{F}$ in the direction given by the zigzag, i.e., in the direction $(M^1 \to M^2)$, $(M^2 \to M^3)$, $(M^3 \to M^4)$, and $(M^4 \to M^1)$, are valid for the WPP.

The set $\mathcal{F}$ can be understood in the following way. Let us consider two edges $e_{13} \in (M^1, M^3)$ and $e_{24} \in (M^2, M^4)$. Then the edges in $E_{\mathcal{C}} \setminus \{e_{13}, e_{24}\}$ can be oriented to obtain a (directed) symmetric graph. Given any such orientation, $\mathcal{F}$ is defined by all the required edges that have been oriented in the direction opposite to the zigzag. In particular, set $\mathcal{F}$ in Figure 1b is defined from the orientation associated with the fractional solution shown in Figure 1a. Note that such a set $\mathcal{F}$ satisfies condition (7). Other sets $\mathcal{F}$ can be defined to obtain valid inequalities but only the one shown in Figure 1b has an associated inequality violated by the fractional solution in Figure 1a.

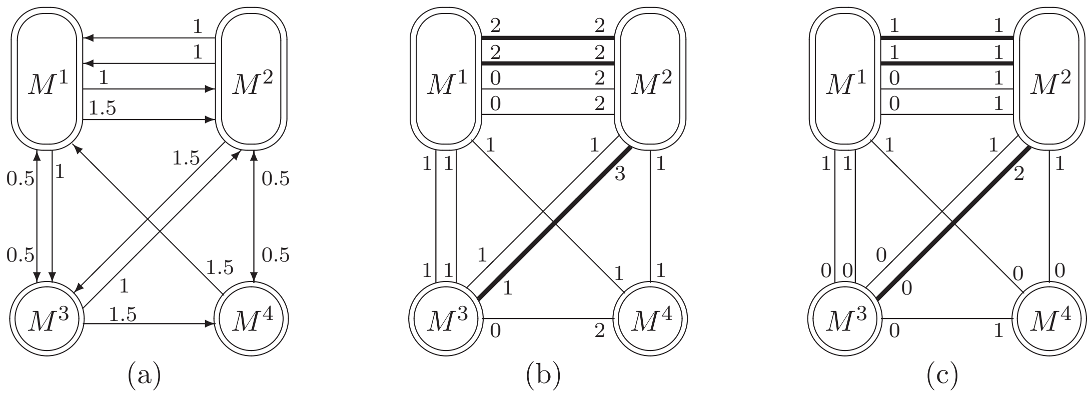


Figure 1: A fractional solution and an odd zigzag configuration with set $\mathcal{F}$ in bold lines.

Like odd-cut inequalities, odd zigzag inequalities can also be written in sparse form. To illustrate this, consider the inequality (8) represented in Figure 1b. Due to the symmetry equation associated with node M^1, the term $x(V \setminus M^1 : M^1)$ in the inequality can be replaced by $x(M^1 : V \setminus M^1)$ to obtain an equivalent inequality. Proceeding in a similar way with node M^2 and then dividing by two, we obtain the equivalent inequality (whose coefficients are shown in

Figure 1c) requiring

$$
\begin{aligned}
& x(M^1 \cup M^2 : M^3 \cup M^4) + x(M^2 : M^1) + x(M^4 : M^3) + x(F_{zz}) \\
\geq\ & \frac{1}{2}\big(|(M^1, M^3)| + |(M^2, M^4)| + |(M^1, M^4)| + |(M^2, M^3)|\big) + |\mathcal{H} \cap \mathcal{F}| + 1,
\end{aligned} \tag{9}
$$

for each $x \in \text{WPP}(G)$.

3 Small WPP polyhedra and new zigzag inequalities

In studying the separation of odd zigzag inequalities, we found some fractional solutions having four edges with value 1.5 forming a non-directed path joining two odd nodes with two even nodes (such as those in Figures 4a, 4b, 5a, and 6a). Given that these fractional solutions cannot be separated by an odd zigzag inequality, we looked for similar classes of facet-inducing inequalities. In order to find them, we computed the complete linear description of the polyhedra associated with the WPP defined on graphs with four vertices and up to ten edges.

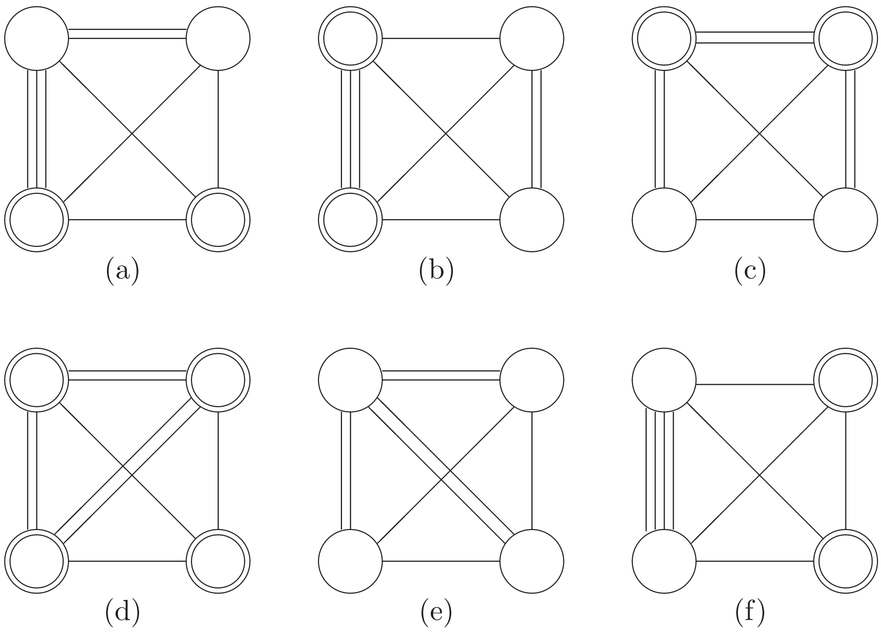


Figure 2: Graphs with four nodes and nine edges

We first considered the complete graph with six edges. Then we added edges to obtain all the possible graphs with seven, eight, nine, and ten edges, respectively. For each graph, the full description of its associated WPP polyhedron, WPP(G), was obtained using PORTA [5]. Trivial and traversing facets appear in the description of all the polyhedra and we do not mention them in what follows. Graphs with seven edges are fully described by odd-cut inequalities, while graphs with six and eight edges are described by odd-cut and odd zigzag inequalities. All non-isomorphic graphs with nine edges are depicted in Figure 2, where odd-degree nodes are represented by a double circle. The polyhedra associated with graphs in Figures 2a, 2b, and 2c are fully described by odd-cut inequalities; for the graph in Figure 2d, odd-cut and odd zigzag inequalities are needed; and for the graph in Figure 2e, no more inequalities are needed. However, the description of the polyhedron associated with the graph in Figure 2f (and Figure 3a) contains new facet-defining inequalities that do not correspond to any of the classes previously mentioned. Similarly, among the ten different graphs with four vertices and ten edges, only the description of the polyhedron associated with the graph shown in Figure 3b contains new facet-defining inequalities.

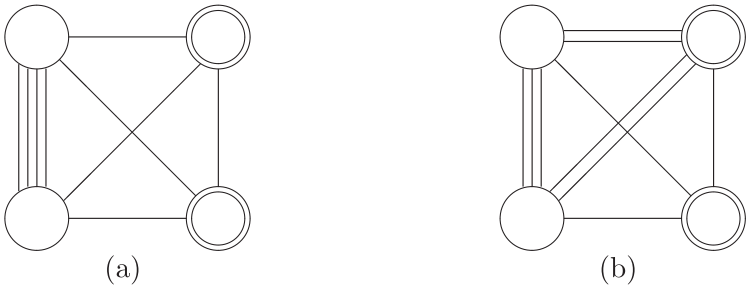


Figure 3: WPP instances whose polyhedra have new facet-inducing inequalities.

Hence, new facet-inducing inequalities were found only for the instances shown in Figure 3. For the instance shown in Figure 3a, the two fractional solutions in Figure 4 can be obtained. Two similar fractional solutions can also be found for the instance depicted in Figure 3b. Note that in Figure 4b the arrangement of the nodes has been changed in order to better illustrate the inequalities that will be described in the following sections. One of the two variables associated with each of the edges $\{1,2\}$, $\{2,3\}$, $\{3,4\}$, and $\{4,1\}$ has value 1.5 in these solutions. In the solution shown in Figure 4a, these edges are traversed in directions $(1,2)$, $(2,3)$, $(4,3)$, and $(1,4)$, while in the solution depicted in Figure 4b they are traversed in the directions $(1,2)$, $(2,3)$, $(3,4)$, and $(1,4)$. Each of these two solutions violates an inequality from the class described in the following. Each type of inequality is determined by the position of the two odd-degree nodes and by the traversal of the four edges $\{1,2\}$, $\{2,3\}$, $\{3,4\}$, and $\{4,1\}$. Note that the four edges with value 1.5 do not form a directed path. For the sake of simplicity, for each type of inequality, we will say that an edge is oriented "in the direction of the zigzag" if it is traversed as described above.

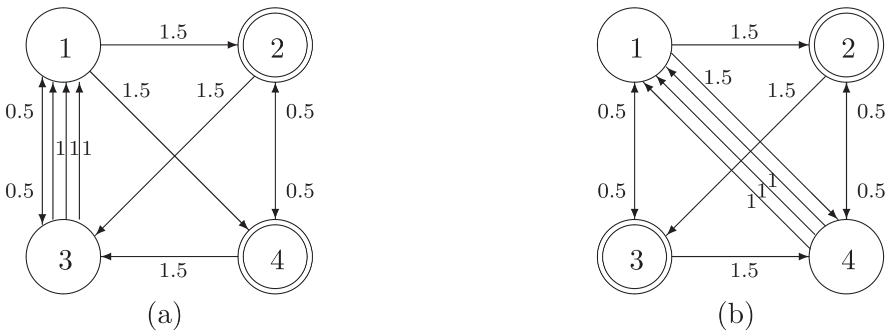


Figure 4: Fractional solutions

3.1 Even-even zigzag inequalities

This class of valid inequalities is violated by fractional solutions similar to those shown in Figures 4a and 5a. The name of the inequalities refers to the degree of the two shores of the edge cutset $\delta(M^1 \cup M^3)$. In this case, the number of edges in this cutset must be even.

Consider a partition of the set of vertices V into four parts, M^1, M^2, M^3, and M^4, where M^2 and M^4 contain an odd number of odd vertices while M^1 and M^3 contain an even number of odd vertices. Let us define a subset of edges $\mathcal{F} \subset \delta(M^1 \cup M^3)$ satisfying

$$\left|(M^1, M^2 \cup M^4) \setminus \mathcal{F}\right| + \left|(M^2 \cup M^4, M^3) \cap \mathcal{F}\right| = \left|(M^1, M^2 \cup M^4) \cap \mathcal{F}\right| + \left|(M^2 \cup M^4, M^3) \setminus \mathcal{F}\right| \quad (10)$$

(see Figure 5b, where edges in $\mathcal{F}$ are represented in bold lines).

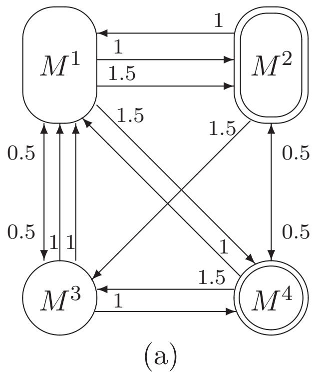
The configuration graph $G_{\mathcal{C}}$ associated with the even-even zigzag inequalities is defined by the partition of V and the set $\mathcal{F}$ above, and by the following pair of coefficients associated with each edge $e_{ij} \in E_{\mathcal{C}}$ (see Figure 5b):

$$(\alpha_{ij}, \alpha_{ji}) = \begin{cases} (0,2), & \forall e_{ij} \in (M^1, M^2 \cup M^4) \setminus \mathcal{F}, i \in M^1, j \in M^2 \cup M^4, \\ (2,2), & \forall e_{ij} \in (M^1, M^2 \cup M^4) \cap \mathcal{F}, \forall e_{ij} \in (M^2, M^4), \\ (1,3), & \forall e_{ij} \in (M^3, M^2 \cup M^4) \cap \mathcal{F}, i \in M^3, j \in M^2 \cup M^4, \\ (1,1), & \text{otherwise.} \end{cases}$$

Note that the pairs with different coefficients correspond to the edges in the left hand side of (10). The corresponding even-even zigzag inequality then requires

$$\begin{aligned} & x(\delta(M^3)) + 2x(M^2, M^4) + 2x(M^2 : M^1) + 2x(M^4 : M^1) + 2x(F_{zz}) \\ \geq\ & \left|(M^1, M^3)\right| + 2\left|(M^2, M^4)\right| + \left|(M^2, M^3)\right| + \left|(M^3, M^4)\right| + 2\left|\delta(M^1) \cap \mathcal{F}\right| + 2 \end{aligned} \quad (11)$$

for all $x \in \text{WPP}(G)$, where $x(F_{zz})$ denotes the variables associated with the edges in $\mathcal{F}$ in the direction of the zigzag, i.e., $(M^1 \to M^2)$, $(M^2 \to M^3)$, $(M^4 \to M^3)$, and $(M^1 \to M^4)$.



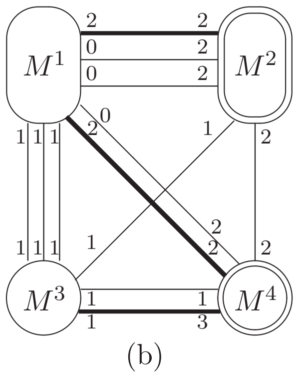


Figure 5: A fractional solution and an even-even zigzag configuration with set $\mathcal{F}$ in bold lines.

The set $\mathcal{F}$ can be chosen in the following way. Let the edges in $E_{\mathcal{C}}$, with the exception of two given edges e_{13}, $f_{13} \in (M^1, M^3)$, and a given edge $e_{24} \in (M^2, M^4)$, be oriented to obtain a (directed) symmetric graph. Given any such orientation, $\mathcal{F}$ is defined to be the set of all the edges that have been oriented in the opposite direction to the zigzag. In particular, the set $\mathcal{F}$ in Figure 5a is defined from the orientation associated with the fractional solution. Note that such a set $\mathcal{F}$ satisfies (10).

Theorem 1 *Even-even zigzag inequalities (11) are valid for WPP(G) for any choice of sets M^i and $\mathcal{F}$ satisfying the above conditions.*

Proof: Let $(\alpha, \beta) \in \mathbb{Z}^{2|E|+1}$ denote the vector of coefficients and the right-hand side of the inequality and let x be a WPP tour. In what follows, given a vector $y \in \mathbb{R}^{2|E|}$, we will call

α-cost of y to αy. Because each WPP tour must traverse every edge in one of its two possible directions, x has at least an α-cost of $\beta - 2$. We can suppose that x traverses all the edges in the left hand side of (10) in the direction corresponding to their lower coefficient, i.e., from $M^1 \cup M^3$ to $M^2 \cup M^4$ (otherwise we have an extra α-cost of two and we are done). Let k be the number of such edges. Since the nodes M^2 and M^4 are odd, the tour x cannot traverse each edge exactly once and an extra matching on these two nodes is needed. Note that all these matchings have α-cost of at least two (and hence we are done), with the exception of the matching with α-cost zero defined by two edges not in $\mathcal{F}$ traversed from M^1 to M^2 and from M^1 to M^4, respectively. So we can suppose that x traverses at least $k+2$ times from $M^1 \cup M^3$ to $M^2 \cup M^4$ and, hence, it traverses also $k+2$ times from $M^2 \cup M^4$ to $M^1 \cup M^3$. Given that the number of edges in the right hand side of (10) is k, x indicates the traversal of the edges in $M^2 \cup M^4$ to $M^1 \cup M^3$ at least two extra times, with an α-cost of at least two. ♦

Theorem 2 *The even-even zigzag inequalities (11) are facet-inducing for WPP(G) if*

- *(i)* $G_{\mathcal{C}} \setminus \mathcal{F}$ *is a complete graph,*
- *(ii) there are three edges* $e_{13}, f_{13} \in (M^1, M^3)$ *and* $e_{24} \in (M^2, M^4)$*, and*
- *(iii) the edges in* $E_{\mathcal{C}} \setminus \{e_{13}, f_{13}, e_{24}\}$ *can be oriented to induce a (directed) symmetric graph where all the edges in* $\delta(M^1 \cup M^3) \setminus \mathcal{F}$ *are oriented in the direction of the zigzag and all the edges in* $\mathcal{F}$ *are oriented in the opposite direction.*

Proof: We first show that the inequalities are facet-inducing for WPP($G_{\mathcal{C}}$). We need to find twice the number of edges in $E_{\mathcal{C}}$ minus three linearly independent WPP tours satisfying $\alpha x = \beta$. Each tour is a vector $x \in \mathbb{Z}^{2|E_{\mathcal{C}}|}$ with two components x_{ij}, x_{ji} associated with each edge $e = \{i, j\} \in E_{\mathcal{C}}$.

We first select the six components corresponding to the given edges e_{13}, f_{13}, and e_{24}. Let $x^d \in \mathbb{Z}^{2|E_{\mathcal{C}}|}$ denote the incidence vector of the symmetric subgraph induced by the orientation mentioned in the theorem. Note that the six selected components in x^d are zero and that $\alpha x^d = \beta - 6$.

Since $G_C \setminus \mathcal{F}$ is a complete graph, it is possible to select another four components associated with four edges not in $\mathcal{F}$, $e_{12} \in (M^1, M^2)$, $e_{14} \in (M^1, M^4)$, $e_{23} \in (M^2, M^3)$, and $e_{43} \in (M^4, M^3)$, in the direction given by the zigzag. It can be seen that, for each unselected variable x_{ij}, a tour based on vector x^d can be constructed satisfying $\alpha x = \beta$ and such that it uses the variable x_{ij} one more time than x^d plus some of the selected components. Moreover, we can build seven more linearly independent WPP tours from x^d by adding to it only some of the selected components, also satisfying $\alpha x = \beta$. If we subtract x^d from all the tours and arrange them in rows, we obtain a full-rank matrix. Hence the even-even zigzag inequalities are facet-inducing for WPP($G_{\mathcal{C}}$). Due to the lifting theorem of [10], they are also facet-inducing for WPP(G). ♦

Even-even zigzag inequalities can also be written in sparse form. Due to the symmetry equation associated with node M^3, the term $x(V \setminus M^3 : M^3)$ in the inequality can be replaced by $x(M^3 : V \setminus M^3)$ to obtain an equivalent inequality that when divided by two requires

$$\begin{aligned} & x(M^3 : V \setminus M^3) + x(M^2, M^4) + x(M^2 : M^1) + x(M^4 : M^1) + x(F_{zz}) \\ \geq\ & \frac{1}{2}\big(|(M^1, M^3)| + 2|(M^2, M^4)| + |(M^2, M^3)| + |(M^3, M^4)|\big) + |\delta(M^1) \cap \mathcal{F}| + 1, \end{aligned} \tag{12}$$

for each $x \in$ WPP(G).

3.2 Odd-odd zigzag inequalities

In this section, we present another family of inequalities that is violated by fractional solutions like those shown in Figures 4b and 6a. Again, the name of the inequalities refers to the degree of the two shores of the edge cutset $\delta(M^1 \cup M^3)$. In this case, the number of edges in this cutset is odd.

Consider a partition of V into 4 parts, M^1, M^2, M^3, and M^4, where M^2 and M^3 contain an odd number of odd vertices while M^1 and M^4 contain an even number of odd vertices. Note that $\delta(M^1 \cup M^3)$ contains an odd number of edges. Let us define a subset of edges $\mathcal{F} \subset \delta(M^1 \cup M^3)$ satisfying

$$\begin{aligned} & |(M^1, M^2 \cup M^4) \setminus \mathcal{F}| + |(M^3, M^4) \setminus \mathcal{F}| + |(M^2, M^3) \cap \mathcal{F}| + 1 \\ = \ & |(M^1, M^2 \cup M^4) \cap \mathcal{F}| + |(M^3, M^4) \cap \mathcal{F}| + |(M^2, M^3) \setminus \mathcal{F}| \end{aligned} \tag{13}$$

(see Figure 6b, where edges in $\mathcal{F}$ are represented in bold lines).

The configuration graph $G_{\mathcal{C}}$ associated with the odd-odd zigzag inequalities is defined by the partition of V and the set $\mathcal{F}$ above, and by the following pair of coefficients associated with each edge $e_{ij} \in E_{\mathcal{C}}$ (see Figure 6b):

$$(\alpha_{ij}, \alpha_{ji}) = \begin{cases} (0,1), & \forall e_{ij} \in (M^1, M^2 \cup M^4) \setminus \mathcal{F}, i \in M^1, j \in M^2 \cup M^4, \\ (0,1), & \forall e_{ij} \in (M^3, M^4) \setminus \mathcal{F}, i \in M^3, j \in M^4, \\ (2,1), & \forall e_{ij} \in (M^2, M^3) \cap \mathcal{F}, i \in M^2, j \in M^3, \\ (1,1), & \text{otherwise}. \end{cases}$$

Again, the pairs with different coefficients correspond to the edges of the left hand side of (13). The corresponding odd-odd zigzag inequality is then

$$\begin{aligned} & x(M^2 \cup M^4 : M^1 \cup M^3) + x(M^1, M^3) + x(M^2, M^4) + x(M^3 : M^2) + x(F_{zz}) \\ \geq \ & |(M^1, M^3)| + |(M^2, M^4)| + |(M^2, M^3) \setminus \mathcal{F}| + |\mathcal{F}| + 1 \end{aligned} \tag{14}$$

for all $x \in \mathrm{WPP}(G)$, where $x(F_{zz})$ denotes the variables associated with the edges in $\mathcal{F}$ in the direction of the zigzag, i.e., $(M^1 \to M^2)$, $(M^2 \to M^3)$, $(M^3 \to M^4)$, and $(M^1 \to M^4)$.

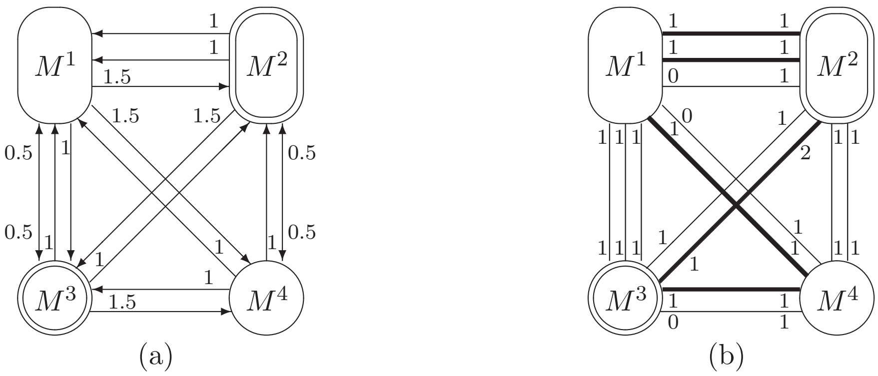


Figure 6: A fractional solution and an odd-odd zigzag configuration with set $\mathcal{F}$ in bold lines.

The following two theorems are given without proofs because they can be proved in an analogous way to Theorems 1 and 2.

Theorem 3 *Odd-odd zigzag inequalities (14) are valid for WPP(G) for any choice of sets M^i and $\mathcal{F}$ satisfying the above conditions.*

Theorem 4 *Odd-odd zigzag inequalities (14) are facet-inducing for WPP(G) if*

(i) $G_{\mathcal{C}} \setminus \mathcal{F}$ *is a complete graph,*

(ii) there are three edges $e_{13} \in (M^1, M^3)$, $e_{24} \in (M^2, M^4)$, *and* $e_{14} \in (M^1, M^4) \cap \mathcal{F}$, *and*

(iii) the edges in $E_{\mathcal{C}} \setminus \{e_{13}, f_{24}, e_{14}\}$ *can be oriented to induce a (directed) symmetric graph where all the edges in* $\delta(M^1 \cup M^3) \setminus \mathcal{F}$ *are oriented in the direction of the zigzag and all the edges in* $\mathcal{F}$ *are oriented in the opposite direction.*

4 Zigzag inequalities and mod-k inequalities

In this section, we prove that zigzag inequalities are mod-4 cuts. A mod-k cut (see [4]) is a rank-1 Chvátal inequality in which the multipliers are restricted to the set $\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}\}$.

Proposition 2 *Odd, even-even, and odd-odd zigzag inequalities are mod-4 cuts.*

Proof: Let us begin with odd zigzag inequalities. Consider the odd zigzag configuration and add the following equations and inequalities multiplied by $\frac{1}{2}$:

– The odd-cut inequality associated with M^1
$x(M^1 : V \setminus M^1) \geq \frac{1}{2}(|(M^1, M^2)| + |(M^1, M^4)| + |(M^1, M^3)| + 1)$,

– the odd-cut inequality associated with $V \setminus M^3$
$x(V \setminus M^3 : M^3) \geq \frac{1}{2}(|(M^1, M^3)| + |(M^2, M^3)| + |(M^3, M^4)| + 1)$,

– the traversing inequalities corresponding to the edges in (M^2, M^4)
$x_{ij} + x_{ji} \geq 1,\ \forall \{i, j\} \in (M^2, M^4)$,

– the traversing inequalities corresponding to the edges in $\mathcal{F}$
$x_{ij} + x_{ji} \geq 1,\ \forall \{i, j\} \in \mathcal{F}$,

– the trivial inequalities corresponding to the variables associated with the edges in $\mathcal{F}$ in the direction given by the zigzag,

– the trivial inequalities corresponding to the variables associated with the edges in $(M^1 \cup M^3, M^2 \cup M^4) \setminus \mathcal{F}$ in the opposite direction to the zigzag, and

– the symmetry equation corresponding to M^2
$x(M^2 : V \setminus M^2) - x(V \setminus M^2 : M^2) = 0$.

We obtain an inequality with integer coefficients and right hand side

$$\frac{1}{2} \cdot \Big(\tfrac{1}{2} \left(|(M^1, M^2)| + |(M^1, M^4)| + |(M^1, M^3)| + 1 \right) + \tfrac{1}{2} \left(|(M^1, M^3)| + |(M^2, M^3)| + |(M^3, M^4)| + 1 \right) + |(M^2, M^4)| + |\mathcal{F}| \Big).$$

From condition (7), it can be seen that this right hand side is equal to

$$\frac{1}{2}\Big(|(M^1, M^3)| + |(M^2, M^4)| + |(M^1, M^4)| + |(M^2, M^3)| + 2|\mathcal{H} \cap \mathcal{F}| + 1 \Big),$$

and, since the left hand side is integer and $|(M^1, M^3)| + |(M^2, M^4)| + |(M^1, M^4)| + |(M^2, M^3)|$ is an even number, the right hand side can be rounded up to

$$\frac{1}{2}\Big(|(M^1, M^3)| + |(M^2, M^4)| + |(M^1, M^4)| + |(M^2, M^3)| + 2|\mathcal{H} \cap \mathcal{F}| + 2\Big),$$

which is exactly the odd zigzag inequality (9) in sparse form.

To obtain the even-even zigzag inequalities in sparse form (12), we add the following inequalities multiplied by $\frac{1}{2}$: the odd-cut inequalities associated with M^2 and with M^4, the traversing inequalities associated with edges in (M^1, M^3), (M^2, M^4), and $\mathcal{F}$, the trivial inequalities corresponding to the variables associated with the edges in $\mathcal{F}$ in the direction given by the zigzag (here, $(M^1 \to M^2)$, $(M^2 \to M^3)$, $(M^4 \to M^3)$, and $(M^1 \to M^4)$), the trivial inequalities corresponding to the variables associated with the edges in $(M^1 \cup M^3, M^2 \cup M^4) \setminus \mathcal{F}$ in the opposite direction to the zigzag, and the symmetry equation corresponding to set M^3 and then we round up the right hand side.

The odd-odd zigzag inequality (14) is obtained by adding the following inequalities multiplied by $\frac{1}{2}$: the odd-cut inequalities associated with M^2, with $V \setminus M^3$, and with set $M^3 \cup M^4$, the traversing inequalities associated with edges in (M^1, M^3), (M^2, M^4), and $\mathcal{F}$, the trivial inequalities corresponding to the variables associated with the edges in $\mathcal{F}$ in the direction given by the zigzag (here, $(M^1 \to M^2)$, $(M^2 \to M^3)$, $(M^3 \to M^4)$, and $(M^1 \to M^4)$), and the trivial inequalities corresponding to the variables associated with the edges in $(M^1 \cup M^3, M^2 \cup M^4) \setminus \mathcal{F}$ in the opposite direction to the zigzag. ♦

Note that, since the odd-cut inequalities are mod-2 inequalities with respect to the original formulation (1)-(4), zigzag inequalities are mod-4 inequalities or, put in another way, mod-2 inequalities of rank two.

5 Branch and cut for the WPP

We have implemented a branch-and-cut algorithm for the WPP in which, besides the well-known heuristic and exact procedures for separating violated odd-cut inequalities (see, e.g., [8]), new separation algorithms for zigzag inequalities are incorporated. It is not known whether or not the problem of separating an arbitrary vector from the windy postman polyhedron using zigzag inequalities is polynomially solvable, but we conjecture that this problem is NP-hard. Hence, we have designed heuristic algorithms for identifying violated odd, even-even, and odd-odd zigzag inequalities. Moreover, a polynomial time algorithm for identifying maximally violated mod-k inequalities has been added, as well as a heuristic algorithm for producing feasible solutions from the fractional LP solutions.

5.1 Zigzag separation procedures

First procedure

This algorithm is designed to separate a fractional solution $x^* \in \mathbb{R}^{2|E|}$ similar to those shown in Figures 1a, 5a, and 6a. For the sake of simplicity, we suppose here that most of the components of x^* are integral, with the exception of some pairs with values $(1.5, 0)$ and $(0.5, 0.5)$. Figure 7 shows a directed graph associated with such a fractional solution in which the number next to an arc $\{i, j\}$ denotes the value of its corresponding variable x^*_{ij}.

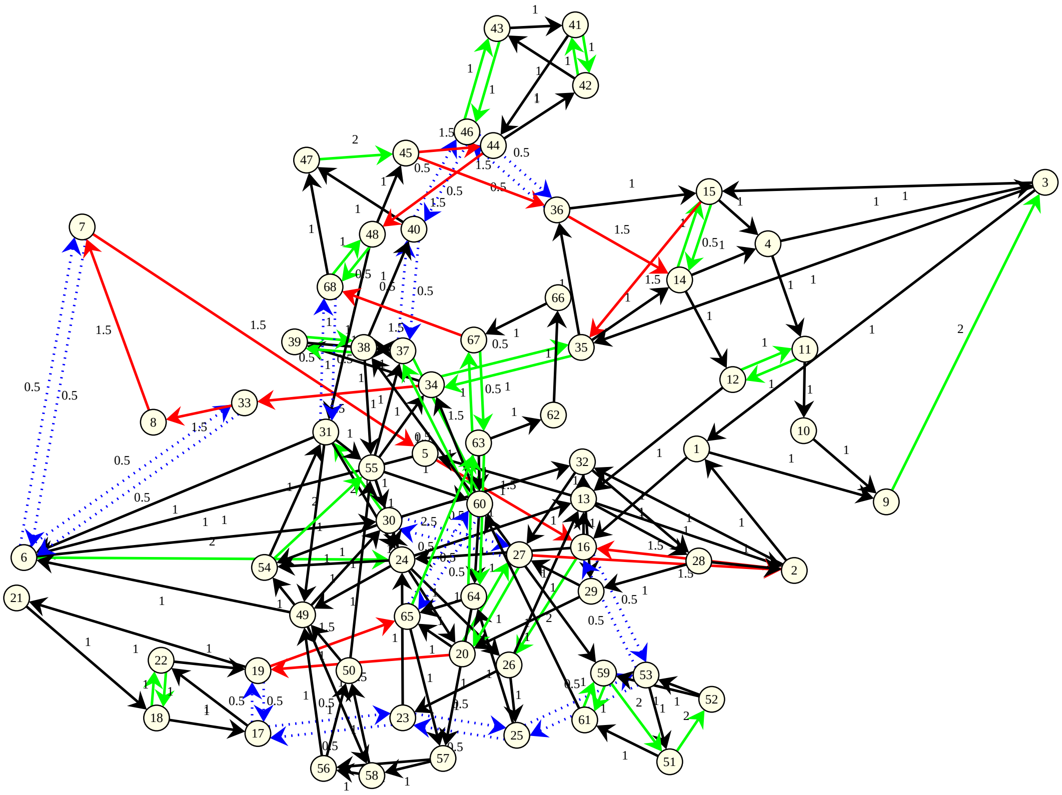


Figure 7: Fractional solution.

We are looking for a zigzag configuration such that the corresponding inequality is violated by x^*. Zigzag inequalities associated with a configuration that has an edge $\{i, j\}$ linking different sets M^k and satisfying $x^*_{ij} + x^*_{ji} \geq 2$ are, according to our experience, typically not violated. Hence, we require first that each such edge belongs to one of the sets M^k. To accomplish this, nodes i and j are shrunk for all such edges in a first step. Moreover, edges $\{i, j\}$ traversed exactly once, i.e., satisfying $x^*_{ij} = 1$ and $x^*_{ji} = 0$ or $x^*_{ij} = 0$ and $x^*_{ji} = 1$, do not seem to affect the violation of the inequality and are therefore deleted. All the resulting isolated nodes are also removed.

At this point we should have one (or more) subgraphs like the one depicted in Figure 8 containing only the edges $\{i, j\}$ satisfying $(x^*_{ij}, x^*_{ji}) = (1.5, 0)$, depicted as arcs in solid lines, or $(x^*_{ij}, x^*_{ji}) = (0.5, 0.5)$ represented by two opposite arcs in dotted lines. For each of these subgraphs we proceed as follows. Each node incident with just two edges is iteratively shrunk into one of its adjacent nodes. We then also shrink all the pairs of nodes linked by two parallel edges. Figure 9a shows the graph obtained after applying this procedure to the one in Figure 8.

After the above procedure, we have a cycle formed by edges with values $(1.5, 0)$ whose nodes are linked by edges with values $(0.5, 0.5)$, which we call *chords* (see Figure 9a). Note that each chord divides the cycle into two parts. We iteratively select two chords and label their end-nodes as *seeds* (since they are the seeds for sets M^k). We check if these two chords cross each other, i.e., each of the two halves defined by a chord contains exactly one of the seeds associated with the other chord. If so, the remaining chords are studied. For each one of these chords, if one of the two halves defined by it contains three seeds then we shrink the other half (which contains just one seed). If this can be done for all the chords, we obtain a graph like the one shown in Figure 9b, which corresponds to a zigzag configuration. However, if a chord defines halves

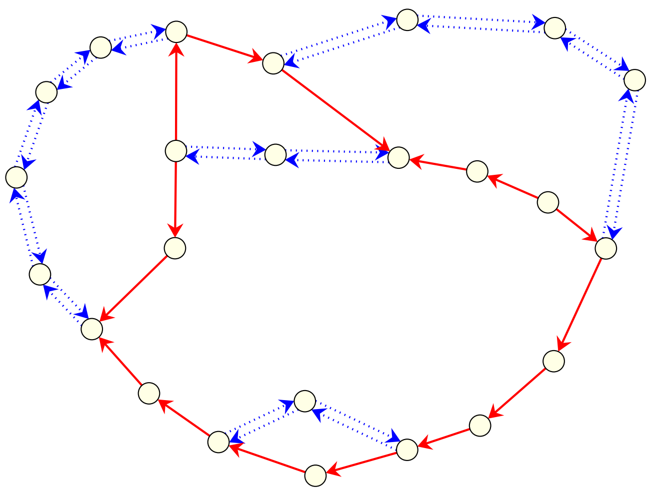
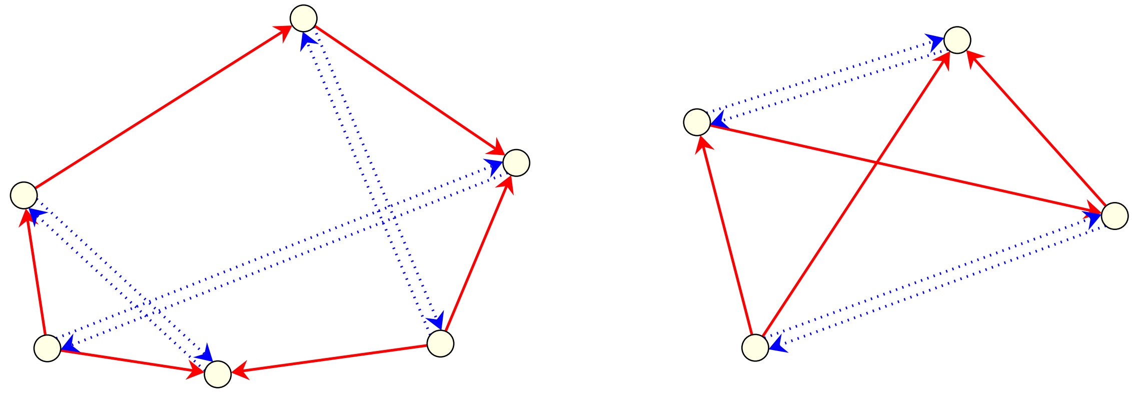


Figure 8: Shrunk fractional solution.

with two seeds each, the procedure fails and we proceed with selecting another pair of chords. Finally, the zigzag configuration on the original graph is built and its corresponding inequality is checked for violation by x^*.



(a) (b)

Figure 9: Cycle with chords and an odd-odd zigzag configuration.

Since zigzag inequalities can also be violated by solutions having fractional values different from 1.5 and 0.5, the above procedure is applied to any fractional solution $x^* \in \mathbb{R}^{2|E|}$ having fractional values as follows:

1) Edges $\{i, j\}$ such that $x^*_{ij} + x^*_{ji} > 1$ are shrunk, with the exception of those described in 2),

2) edges $\{i, j\}$ such that $1.25 < x^*_{ij} + x^*_{ji} < 1.75$ and $x^*_{ij} < 0.2$ or $x^*_{ji} < 0.2$ are considered as satisfying $(x^*_{ij}, x^*_{ji}) = (1.5, 0)$ or viceversa,

3) for the remaining edges, $x^*_{ij} + x^*_{ji} = 1$, those with $x^*_{ji} < 0.1$ are considered as satisfying $(x^*_{ij}, x^*_{ji}) = (1, 0)$ and are deleted and the ones with $0.1 < x^*_{ij} < 0.4$ or $0.1 < x^*_{ji} < 0.4$ are shrunk, and

4) all the other edges are treated as satisfying $(x^*_{ij}, x^*_{ji}) = (0.5, 0.5)$.

Second procedure

Unlike the previous algorithm, which is capable of finding odd, even-even, and odd-odd violated zigzag inequalities, the procedure described in this section is designed to separate only odd zigzag inequalities. The idea of the algorithm is that the fractional solutions similar to the one shown in Figure 1a satisfy the odd-cut inequalities corresponding to the (odd) sets M^k, $k = 1, 2, 3, 4$, with equality. It works as follows.

During the execution of the branch-and-cut algorithm, all the odd-cut inequalities found are stored. We consider all the stored inequalities satisfied by x^* with equality. We then select shores $S \subset V$ of these inequalities such that all pairs of variables associated with edges in $(S, V \setminus S)$ have values $(1, 0)$, with the exception of the variables associated with the three edges that have values $(1.5, 0)$, $(0, 1.5)$, and $(0.5, 0.5)$ respectively. Three such sets are iteratively selected. If they have no nodes in common, these three sets and the rest of V are the candidates to be the sets M^k of the odd zigzag configuration. Finally, if the sets can be arranged in such a way that the arcs defined by the variables with value 1.5 form a directed cycle, then we have an odd zigzag configuration, whose associated inequality is checked for violation by x^*.

5.2 Separation of mod-k inequalities

In addition to these problem-specific inequalities we have also incorporated the separation of more general maximally violated mod-k inequalities according to procedures described in [3, 4]. The basic work consists of solving a system of congruences where the system is composed of all the binding inequalities of the last LP plus an inequality for each symmetry equation.

In our algorithm we have used the implementation from [16] for the separation of maximally violated mod-2 inequalities. We apply the separation procedure to a system including not only inequalities of the original formulation, but also others that have been added as cutting planes. Therefore, the inequalities generated are not only mod-2 cuts, but could be mod-k for arbitrary k, since they would be mod-2 with respect to a formulation that already includes other mod-k inequalities. Given that zigzag inequalities are mod-4 inequalities, it is likely that some of the maximally violated mod-k inequalities are zigzag inequalities that, as they are facet-inducing, may be useful for the branch-and-cut algorithm.

The system of congruences to be solved for the separation of mod-k inqualities usually has very many solutions (possibly leading to the same inequalities). Therefore an important practical issue is the selection of cuts to be added to the LP. A nice description of different strategies can be found in [19]. We have basically used the one described in [16].

Because mod-k separation is computationally very expensive and we have faster separation algorithms for zigzag and odd-cut inequalities (which are mod-2 inequalities), we call this routine only at the root node.

5.3 Initial relaxation and cutting-plane algorithm

The initial LP relaxation contains the odd-cut inequalities associated with the odd-degree vertices and with the connected components of the subgraph of G induced by the odd-degree vertices, if $|S|$ is odd. At each iteration of the cutting-plane algorithm the separation procedures are called in the following order:

1) Odd-cut inequalities separation heuristics,

2) when no violated odd-cut inequalities are found in step 1, odd-cut inequalities exact separation algorithm is called,

3) when no violated odd-cut inequalities are found in steps 1 and 2, zigzag inequalities separation procedures are executed, and

4) (only at the root node) when no violated inequalities are found in steps 1–3, mod-k cuts separation procedure is applied.

If violated inequalities are found in any of the steps 1–4, they are added to the current LP and, after it is solved, the loop starts from step 1 again. However, the mod-k cuts separation procedure is not allowed to be executed in two consecutive iterations of the loop, so, if after adding violated mod-k cuts inequalities no more violated odd-cut or zigzag inequalities are found in steps 1–3, the loop terminates.

The cutting-plane procedure is applied at each node of the branch-and-cut tree until no new violated inequalities are found or a stopping criterium, called *tailing-off*, is satisfied. In our implementation the cutting-plane generation stops when the increase in the objective function during the last five iterations is less than 0.0004%. At the root node this percentage has been set to 0.0002%.

5.4 Upper bounds

In order to get good upper bounds that decrease the size of the search tree, a heuristic algorithm based on the information given by the fractional solution x^* obtained after processing each node has been implemented.

The algorithm first builds a directed multigraph G^* associated with x^* in the following way. G^* contains exactly $\lfloor x^*_{ij} \rfloor$ copies of an arc (i,j) when $x^*_{ij} \geq 1$, and one copy of arc (i,j) when $0.75 \leq x^*_{i,j} < 1$. In addition, for each edge $\{i,j\}$ such that x^*_{ij} and x^*_{ji} are less than 0.75, we add one arc (i,j) or (j,i) to G^*. We add the arc (i,j) if $x^*_{ij} > 0.5$ and $x^*_{ji} \leq 0.5$ or the arc (j,i) if $x^*_{ji} > 0.5$ and $x^*_{ij} \leq 0.5$. Otherwise we add the arc associated with the direction of lower cost.

Consider now the directed graph G_{aux} resulting from replacing each edge in the original graph G by two opposite arcs. We assign infinite capacity and weights c_{ij}, c_{ji} to these arcs. We add to G_{aux} one arc (j,i) for those pair of nodes i,j such that exactly one copy of arc (i,j) and no copy of arc (j,i) exist in G^*. These arcs are usually called *artificial arcs* (since they represent the possibility of reorienting its corresponding opposite arc) and have capacity two and weight $\frac{1}{2}(c_{ji} - c_{ij})$. Now a minimum cost flow (f_{ij}) is computed on G_{aux} with demands and supplies at each vertex defined by the difference between the number of arcs entering i and leaving i in G^*.

For each arc (i,j) that is not artificial in G_{aux}, f_{ij} additional copies of arc (i,j) are added to G^*. In addition, for each artificial arc (j,i) with non zero flow we proceed as follows:

– If $f_{ji} = 2$, then arc (i,j) in G^* is replaced by its opposite arc (j,i), and,

– if $f_{ij} = 1$, then arc (i,j) in G^* is replaced by an edge $\{i,j\}$.

At this point, G^* can be a mixed graph. We then solve a minimum-cost matching problem defined on a complete graph whose nodes are the ones incident with an odd number of edges in G^*, and the edge costs are computed from the shortest paths in the original graph as follows. Given two nodes i and j, we find the shortest path from i to j and compute the sum of the average costs $\frac{1}{2}(c_{uv} + c_{vu})$ of its edges $\{u, v\}$. The same value is computed for the shortest path from j to i and the minimum of these two values is selected as the cost of the edge linking i and j. Then, for each edge in the optimal matching, we add to G^* a copy of each original edge in the corresponding shortest path. Each edge is oriented in the direction given by its smallest cost and another similar flow problem is solved (see also [21]). This procedure produces a directed graph representing a feasible solution for the WPP. This solution is improved by applying three simple procedures described in [1]. The first two look for cycles in the solution graph such that after deleting them or reversing the direction of their arcs, a better solution is obtained. The third one looks for directed paths that can be replaced by a shortest path to obtain a better solution.

6 Computational experiments

We present here the computational results obtained on different sets of instances. The branch-and-cut procedure has been coded in C/C++ using the CPLEX 9.0 MIP Solver with Concert Technology 2.0. Default settings for CPLEX were not used. Specifically, CPLEX presolve and heuristic algorithms and cut generation are turned off, the optimality gap tolerance is set to zero, and strong branching and depth-first search are selected. Tests were run on an Intel Pentium IV 2.80GHz and 2GB RAM with a time limit of 10 hours. All the data instances are publicly available ([9]).

6.1 Data instances

We have tested the branch-and-cut procedure on large, randomly generated WPP and MCPP instances. Type A instances correspond to pure random graphs, while type B instances are associated with graphs that try to imitate street networks.

Three parameters are considered to generate a WPP instance: the number of vertices n, $n \in \{500, 1000, 1500, 2000, 3000\}$, the average vertex degree d, $d \in \{3, 4, 5, 6\}$, and an integer number a used to generate asymmetric costs, $a \in \{10, 20, 50\}$. First, the vertex set V is constructed by randomly generating n points in a grid of size 1000×1000. For the instances of type A, the edge set E is obtained by randomly generating $n\frac{d}{2}$ pairs of vertices. The cost of edge $\{i, j\}$ is defined to be $c_{ij} = \lfloor b_{ij} + 0.5 \rfloor$, where b_{ij} denotes the Euclidean distance between i and j. If the resulting graph (V, E) is not connected, edges in d edge-disjoint trees spanning the connected components of the graph are also added to E. At this point, we have an undirected graph. In order to obtain asymmetric edge costs, the following strategy is applied. First, the value c_a is computed as the a^{th} percentile of the edge costs ($a \in \{10, 20, 50\}$). Then, for each edge $e = \{i, j\} \in E$, we let k_1 and k_2 be two integer values randomly selected in the interval $[-c_a, c_a]$ and set $c_{ij} = \max\{1, c_{ij} + k_1\}$, and $c_{ji} = \max\{1, c_{ji} + k_2\}$. This strategy is similar to the one proposed in [20] except that, in this last paper, the values chosen for c_a are five, eight, and ten. In this way, we have generated 60 pure random WPP instances named `WA0531` to `WA3065`, where for example "WA" refers to a WPP instance of type A , the first two digits are the number of vertices divided by 100, the third digit is the vertex degree and the last digit is the value of parameter a divided by 10.

For the type B instances we proceed as above, except for the generation of edges. Here, for each vertex $i \in V$, the d edges connecting i to its d closest neighbors are added to E. The idea is to avoid long edges crossing the graph that would not appear in real networks. If the resulting graph is not connected, edges in d edge-disjoint trees spanning the connected components are also added to E. Furthermore, each edge $\{i, j\} \in E$ such that there is a vertex k satisfying $c_{ij} \geq 0.98(c_{ik} + c_{kj})$ is removed from E to avoid "almost parallel" edges. We have obtained 60 WPP instances of type B named `WB0531` to `WB3065`.

Similarly, 60+60 MCPP instances have been generated. The undirected graphs are generated as above, again with $n \in \{500, 1000, 1500, 2000, 3000\}$ and $d \in \{3, 4, 5, 6\}$. To obtain mixed graphs, each edge is transformed into an arc with probability $p \in \{0.25, 0.5, 0.75\}$. When a given edge is decided to be changed into an arc, we choose one of its two possible orientations with probability 0.5. If the resulting graph is not strongly connected, some arcs joining different strongly connected components are changed back into edges. The 120 MCPP instances are named `MA0532` to `MB3067`, where for example "MA" refers to a MCPP instance of type A , the first two digits are the number of vertices divided by 100, the third digit is the vertex degree and the last digit is the first significative digit of parameter p.

6.2 Computational results

Tables 1 and 2 show the characteristics of the instances above and the computational results obtained on them. They contain the problem type and the number of instances of each set, the number of nodes, and the average, minimum, and maximum number of edges (and arcs in the case of mixed instances). In both tables the next column shows the number of instances solved to optimality for each set. For the instances solved within the time limit of 10 hours, columns "Time" and "BC Nodes" present the average computing time (in seconds) and the number of tree nodes explored. The last column shows the average gap between the final lower bound and the best feasible solution found for those unsolved instances for which a feasible solution was found. In four out of 33 unsolved instances (all of them WPPs with 3000 nodes), the algorithm ran out of memory before finishing the computation for the root node (and without execution of the heuristic algorithm).

Set	# of Instances	# of Nodes	# of Edges Aver.	Min	Max	# of Inst. solved to Optimality	Time (sec.)	BC Nodes	Gap (%)
WA05	12	500	1160.5	813	1518	12	10.4	3.7	–
WB05	12	500	1212.8	874	1555	12	13.5	0.6	–
WA10	12	1000	2316.8	1641	3018	12	168.1	27.1	–
WB10	12	1000	2433.9	1743	3110	12	159.5	2.5	–
WA15	12	1500	3493.1	2478	4530	11	1595.3	75.3	0.01
WB15	12	1500	3654.6	2631	4670	12	1457.6	24.3	–
WA20	12	2000	4644.6	3303	6036	12	2203.6	68.7	–
WB20	12	2000	4826.3	3467	6129	8	5700.0	62.7	0.12
WA30	12	3000	6961.3	4986	9066	6	4920.3	225.8	0.01
WB30	12	3000	7140.8	5176	9085	2	15160.6	110.0	0.19

Table 1: Computational results on WPP instances

As can be seen in Table 1, our algorithm solved all but one of the WPP instances up to 1500 nodes. It was also capable of solving to optimality most of the 2000 nodes WPP instances and some of the 3000 nodes ones. On the other hand, instances of type B (generated trying to imitate real networks) proved to be more difficult than those generated completely at random. The results shown in this table also confirm that our algorithm is capable of solving very large WPP instances with up to 3000 nodes and 9000 edges. As far as we know, these are the largest instances ever solved.

Set	# of Edges Aver.	Min	Max	# of Arcs Aver.	Min	Max	# of Inst. solved to Optimality	Time (sec.)	BC Nodes	Gap (%)
MA05	615.1	351	1112	542.4	193	1082	12	3.6	0.7	-
MB05	626.3	320	1131	583.8	202	1174	12	7.7	0.4	-
MA10	1235.3	693	2235	1083.9	340	2253	12	3891.2	139.8	-
MB10	1260.1	636	2293	1182.1	416	2327	11	89.9	4.0	0.01
MA15	1851.7	1011	3423	1627.2	576	3329	11	911.7	38.1	0.02
MB15	1880.2	964	3540	1750.5	615	3475	12	2373.4	22.8	-
MA20	2462.6	1355	4528	2182.8	733	4515	11	751.0	41.4	0.01
MB20	2499.8	1243	4552	2329.3	850	4583	10	1392.1	10.6	0.18
MA30	3704.8	1999	6795	3254.3	1097	6603	8	877.0	15.9	0.09
MB30	3746.3	1992	6799	3384.5	1206	6742	9	685.3	7.0	0.97

Table 2: Computational results on MCPP instances

Table 2 reports about the computational results obtained on the MCPP instances. The number of instances and the number of nodes correspond to those of Table 1. The performance of the branch-and-cut procedure is even better on these types of instances with only 12 out of 120 instances remaining unsolved. The algorithm solved instances with 3000 nodes and up to 9000 arcs and edges to optimality, and the average gap obtained on the unsolved instances was less than 1%. Note that, in this case, instances of type A and B seem to be of similar difficulty. Another interesting observation is that, for a given number of arcs and edges, instances with a similar number of arcs and edges seem to be harder than those with an unequal proportion of arcs or edges.

Set	odd cut	odd zigzag	odd-odd zigzag	even-even zigzag	mod-k
WA05	339.8	0.2	0.5	0.2	8.3
WB05	895.1	12.5	1.9	0.8	0.0
WA10	769.4	0.1	1.2	0.1	4.2
WB10	2327.9	25.1	3.8	0.8	4.2
WA15	1827.8	0.5	0.2	0.8	0.0
WB15	5312.5	35.9	6.3	2.7	4.2
WA20	1833.2	1.3	0.2	0.3	0.0
WB20	7917.6	50.4	5.4	3.8	12.5
WA30	2568.5	0.0	1.0	0.2	0.0
WB30	14584.0	42.5	10.5	7.5	25.0

Table 3: Average number of valid inequalities on WPP instances

The average number of valid inequalities found by the separation algorithms for the optimally solved instances can be seen in Tables 3 and 4. Note that the number of valid inequalities found for the WPP instances is larger than for the MCPP instances. Also, instances of type B present a larger number of cuts than those of type A, which are generated completely at random. As expected from the description of small polyhedra discussed in Section 3, odd zigzag inequalities appear more frequently than the other types of zigzag inequalities. Given that there are no violated odd-cut inequalities when the mod-k inequalities separation algorithm is called, the inequalities in the last column must be either zigzag inequalities or unknown valid inequalities. Note that the number of violated zigzag and mod-k inequalities found is small. These results can be explained by considering that, given that they are computationally demanding algorithms, the zigzag inequalities separation procedures are invoked only when no odd-cut inequalities are found and the mod-k inequalities separation routine only at the end of the root node. Additionally, the use of a tailing-off strategy can reduce the number of times these procedures are called even further.

Set	odd cut	odd zigzag	odd-odd zigzag	even-even zigzag	mod-k
MA05	165.4	0.0	0.1	0.0	4.2
MB05	481.4	6.2	1.0	0.5	0.0
MA10	1714.8	0.6	0.6	0.1	0.0
MB10	1271.0	12.3	3.0	3.1	0.0
MA15	981.2	0.1	0.1	0.5	4.5
MB15	2996.3	29.6	7.7	4.5	12.5
MA20	973.8	0.3	0.9	0.2	4.5
MB20	2901.7	13.1	2.6	2.6	0.0
MA30	883.8	0.9	0.1	0.9	0.0
MB30	2323.4	19.1	2.9	0.9	11.1

Table 4: Average number of valid inequalities on MCPP instances

On the other hand, we want to point out that our algorithm is not significantly faster than the variant that only separates odd-cut inequalities (i.e., basically the approach proposed in [12]). We have compared both versions and, on average, although the number of nodes of the branch-and-cut tree explored by our algorithm is lower, the total computing times are similar. However, we would like to point out that our first research aim was the investigation of the polyhedral structure of WPP polyhedra and the definition of new facet-defining inequalities. Our second aim was to develop separation heuristics for the new inequalities and to assess their usefulness for practical computations. Therefore, in order to study with more detail the effect of zigzag inequalities and mod-k inequalities, we have done some additional experiments.

For these experiments, the cutting-plane algorithm from the branch-and-cut procedure has been run using three different separation strategies and without tailing-off. In the first strategy, only odd-cut inequalities are separated. In the second one, we also use the zigzag inequalities separation procedures every ten iterations or when no violated odd-cut inequalities are found. The last one searches also for violated mod-k inequalities when no other violated inequality is found, allowing the addition of a maximum of 500 mod-k inequalities each time. The results of these experiments are shown in Tables 5 to 8.

Table 5 shows the number of WPP instances solved to optimality with each one of the three strategies. As can be seen, using the separation of zigzag inequalities helps the algorithm solve 29 additional instances to optimality, while mod-k inequalities have contributed to solving to optimality two more instances. On Table 6 the contribution of the zigzag inequalities and mod-k inequalities is measured in terms of the reduction, in percentage, of the gap between the final lower bound and the optimal solution. This average gap has been computed for the instances that could not be solved at the root node with the first strategy. The number of such instances is shown in the column "# of instances". It can be observed that a reduction of the gap of 39.95% is achieved by using zigzag inequalities, while the use of mod-k inequalities provides an additional reduction of about 2.5%. We want to point out that the reduction of the gap using zigzag inequalities in the instances of type B is more than four times the reduction obtained in the instances of type A. This agrees with the results presented in Tables 3 and 4 that show that the number of zigzag inequalities found in instances of type B is larger than in those of type A, which had been generated completely at random.

Set	odd-cut	zigzag	mod-k
WA05	5	6	7
WB05	6	10	10
WA10	4	4	4
WB10	6	9	9
WA15	3	3	3
WB15	2	10	11
WA20	0	2	2
WB20	0	6	6
WA30	0	1	1
WB30	0	4	4
Total	26	55	57

Table 5: Number of WPP instances solved to optimality at the root node

Set	# of Instances	Gap Reduction zigzag	Gap Reduction zigzag+mod-k
WA05	7	16.81%	28.57%
WB05	6	67.03%	67.03%
WA10	8	11.18%	15.63%
WB10	6	75.54%	80.98%
WA15	9	11.33%	11.33%
WB15	10	90.93%	93.40%
WA20	12	10.85%	11.79%
WB20	12	54.94%	54.94%
WA30	12	21.28%	21.28%
WB30	12	39.59%	39.59%
Average		39.95%	42.46%

Table 6: Gap reduction at the root node using zigzag and mod-k inequalities on WPP instances

Tables 7 and 8 show the corresponding results for the MCPP instances. It can be seen that almost half of the instances can be solved at the root node using only odd-cut inequalities, which confirms that instances on mixed graphs are easier than those on "windy" graphs. The contribution of zigzag and mod-k inequalities in terms of the gap reduction is similar to that on "windy" graphs, although its effect on the number of instances solved is less significant, maybe because the number of instances already solved by means of odd-cut inequalities is larger.

From the above results, we can say that the new separation procedures clearly show the potential of the inequalities, although the additional time needed for separation does not lead to a reduction of the total time employed by the branch-and-cut procedure. Faster separation heuristics would be desirable, but we think that the new inequalities increase the chance of solving previously unsolved problems.

Acknowledgments

The authors want to thank the three referees for their careful reading of the manuscript and for their many comments and suggestions that have contributed to improve the paper content and readability. In particular, several remarks regarding the discussion of mod-k inequalities were pointed out by one of the referees.

Set	odd-cut	zigzag	mod-k
MA05	10	10	10
MB05	9	11	11
MA10	6	7	7
MB10	6	8	8
MA15	6	6	6
MB15	6	10	10
MA20	5	5	5
MB20	6	7	7
MA30	5	6	6
MB30	5	6	6
Total	64	76	76

Table 7: Number of MCPP instances solved to optimality at the root node

Set	# of Instances	Gap Reduction zigzag	Gap Reduction zigzag+mod-k
MA05	2	0.00%	39.59%
MB05	3	66.67%	66.67%
MA10	6	21.94%	27.50%
MB10	6	62.22%	62.51%
MA15	6	1.34%	8.79%
MB15	6	95.37%	97.01%
MA20	7	0.05%	1.14%
MB20	6	51.35%	51.35%
MA30	7	14.35%	14.35%
MB30	7	46.87%	46.87%
Average		36.02%	41.58%

Table 8: Gap reduction at the root node using zigzag and mod-k inequalities on MCPP instances

A. Corberán, I. Plana and J.M. Sanchis wish to thank the Ministerio de Educación y Ciencia of Spain (projects MTM2006-14961-C05-02 and MTM2009-14039-C06-02) for its support.

References

[1] E. Benavent, A. Carrotta, A. Corberán, J.M. Sanchis & D. Vigo (2007): "Lower Bounds and Heuristics for the Windy Rural Postman Problem". *European Journal of Operational Research* 176, 855–869.

[2] P. Brucker (1981): "The Chinese Postman Problem for mixed graphs". Proc. Int. Workshop. Lecture Notes in Computer Science 100, 354–366.

[3] A. Caprara & M. Fischetti (1996): "$\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts". *Mathematical Programming* 74, 221–235.

[4] A. Caprara, M. Fischetti & A.N. Letchford (2000): "On the separation of maximally violated mod-k cuts". *Mathematical Programming* 87, 37–56.

[5] T. Christof & A. Loebel (1998): "PORTA – A Polyhedron Representation Algorithm." www.informatik.uni-heidelberg.de/groups/comopt/software/PORTA/.

[6] N. Christofides, E. Benavent, V. Campos, A. Corberán & E. Mota (1984): "An Optimal Method for the Mixed Postman Problem". In P. Thoft-Christensen (Ed.) *System Modelling and Optimization*. Lecture Notes in Control and Information Sciences 59, Springer.

[7] A. Corberán, I. Plana & J.M. Sanchis (2006): "Zigzag inequalities: A new class of facet-inducing inequalities for Arc Routing Problems". *Mathematical Programming* 108, 79–96.

[8] A. Corberán, I. Plana & J.M. Sanchis (2007): "A Branch & Cut Algorithm for the Windy General Routing Problem and special cases". *Networks* 49, 245–257.

[9] A. Corberán, I. Plana & J.M. Sanchis (2007): "Arc Routing Problems: Data Instances." www.uv.es/corberan/instancias.htm.

[10] A. Corberán, I. Plana & J.M. Sanchis (2008): "The Windy General Routing Polyhedron: A Global View of many Known Arc Routing Polyhedra". *SIAM J. Discrete Mathematics* 22, 606–628.

[11] M. Grötschel & Z. Win (1988): "On the Windy Postman Polyhedron". Report No. 75, Schwerpunktprogram der Deutschen Forschungsgemeinschaft, Universität Augsburg, Germany.

[12] M. Grötschel & Z. Win (1992): "A Cutting Plane Algorithm for the Windy Postman Problem". *Mathematical Programming* 55, 339–358.

[13] M. Guan (1984): "On the Windy Postman Problem". *Discrete Applied Mathematics* 9, 41–46.

[14] E. Minieka (1979): "The Chinese Postman Problem for Mixed Networks". *Management Science* 25, 643–648.

[15] D. Naddef & G. Rinaldi (1991): "The Symmetric Traveling Salesman Polytope and its Graphical Relaxation: Composition of Valid Inequalities". *Mathematical Programming* 51, 359–400.

[16] M. Oswald, G. Reinelt & H. Seitz (2009): "Applying mod-k cuts for solving linear ordering problems". *TOP* 17, 158–170.

[17] C.H. Papadimitriou (1976): "On the complexity of edge traversing". *Journal of the Association for Computing Machinery* 23, 544–554.

[18] T.K. Ralphs (1993): "On the Mixed Chinese Postman Problem". *Operations Research Letters* 14, 123-127.

[19] K. Wenger (2004): "Generic Cut Generation Methods for Routing Problems". *PhD Dissertation*, University of Heidelberg, Germany.

[20] Z. Win (1987): "Contributions to Routing Problems". *PhD Dissertation*, University of Augsburg, Germany.

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

[22] F.J. Zaragoza Martínez (2008): "Series-Parallel Graphs are Windy Postman Perfect". *Discrete Mathematics* 308, 1366–1374.