On the Rural Postman Problem: Tight Lower Bounds Based on a New Formulation

Elena Fernández[1], Oscar Meza[2,*], Robert Garfinkel[3], Maruja Ortega[2,*]

[1] Statistics and Operations Research Department, Technical University of Catalunya, Spain.

[2] Computer Science and Information Technology Department, Simón Bolívar University, Apartado 89000, Baruta, Venezuela.

[3] School of Business Administration, University of Connecticut
368 FairField Road, U-41IM, Room 302, Storrs, Connecticut
06269-2041, U.S.A.

e-mail: elena@eio.upc.es, meza@ldc.usb.ve, robg@sba.uconn.edu, mof@ldc.usb.ve

Abstract

In this work we focus on a new integer programming approach for the Rural Postman Problem which is not equivalent to any of the previous models. Previous models have general integer variables and a combination of linear and modular constraints. This contrasts with our formulation which is mixed linear integer programming where the integer variables are binary.

The algorithmic possibilities of the model to provide both sharp lower bounds and good feasible solutions are explored. We propose a cutting plane algorithm based on a very few number of valid inequalities as compared with previous works and resulting in very sharp lower bounds.

Different types of inequalities are valid for the considered model. We include matching inequalities, connectivity inequalities and K-C inequalities. These inequalities have been used as cutting planes to strenghthen the LP relaxation of our model. Most of the test problems have been optimally solved and problems where optimality is not achieved the gap with respect to the best known solution is extremely small as compared with previous results.

key words: Optimization, heuristics, GRASP, Combinatorial Optimization, graphs.

[*] This research has been partially supported by the Spanish Agency of Iberoamerican Cooperation (A.E.C.I.)

1. Introduction

In [5], Garfinkel and Webb propose a mixed integer programming model for the Rural Postman Problem (RPP), where all the integer variables are binary. In this formulation, every feasible solution is a solution to the RPP, but the contrary is not true. However, there is at least one optimal solution in the set of feasible solutions of the model.

In 1981 Christofides et al. [3] proposed an integer programming model for the RPP, that will be referred to as *the classic model*. To the knowledge of the authors, all previous work on RPP is based on the classic model. The main differences between the classic model and the model proposed by Garfinkel and Webb [5] are: first, the set of feasible solutions to Garfinkel and Webb's model is a subset of the set of all feasible solutions to the RPP; second, the number of constraints in the classic model is exponential, while in Garfinkel and Webb's model, it is a linear function on the number of nodes of the graph; and third, Garfinkel and Webb's model reduces to find a perfect matching satisfying certain conditions, while the classical model maintains the original nature of the problem.

Recent research on the RPP has followed mainly the following three directions:

a) Generate valid inequalities and facets of the convex hull of solutions of the RPP.(see, for instance, [1], [2] and [6]).
b) Use of results from (a) to design cutting plane or branch and cut algorithms . (see, for instance, [1], [2], [3] and [6]).
c) Design of approximation algorithms for the RPP (see, for instance, [4] and [8])

Our first interest has been to explore the algorithmic possibilities of the Garfinkel and Webb's model to provide both sharp lower bounds and good feasible solutions for the RPP. We propose a cutting plane algorithm which works extremely well as compared with the best known results [2].

This paper is organized as follows: in order to introduce concepts and terminology, in Section 2 we present the Corberán's model, a compact version of the classical model, and we propose a new model, the RPP(GR) model, based on Garfinkel and Webb's model, where the number of variables is considerably reduced. Section 3 presents new valid inequalities that can be used in separation algorithms. A cutting plane algorithm based on RPP(GR) model is presented in Section 4. In Section 5 we discuss computational results. Finally, the conclusions of the work are presented in section 6 and references in section 7.

2. A Mixed Integer Programming Model for the Rural Postman Problem.

The Rural Postman Problem (RPP) is stated as following:
Let $G=(V,E)$ be a connected undirected graph, without multiple edges. Let $C: E \rightarrow \Re$ be a cost function on the set of edges E. Let $E_R \subseteq E$ be a subset of edges of E, called "the set of required edges".

RPP consists on finding a least cost closed path in G that includes, at least once, each edge in E_R. By closed path we mean an alternate sequence $<v_0,e_1,v_1,e_2,v_2, \ldots ,e_n,v_n>$ where each v_i is a node, each e_i is an edge, $v_0=v_n$, and the endpoints of each e_i are v_{i-1} and v_i. The cost of a path is the sum of the costs of the edges in it. If an edge appears more than once in a path, its cost will be added to the cost of the path as many times as the edge appears in it.

The RPP has been modeled as and linear integer program by Christofides et al. [3], Corberán et al [2] and Ghiani and Laporte [6]. These models have an exponential number of constraints and present the difficulty of having "modulus" constraints.

In their work [5] Garfinkel and Webb propose a new definition for the RPP:

Definition: Let $G_R=(V_R,E_R)$ be a non oriented and not necessarily connected graph, where each connected component contains at least one edge. Let $E_D=\{\{i,j\}/i\in V_R$ and $j\in V_R$, $i\neq j\}$, with costs (or distances) d_{ij} associated to each $\{i,j\}$. Find a multiset (a collection of not necessarily distinct elements) E_P of edges in E_D, with minimum total cost, such that $G_{R\cup P}=(V_R, E_R\cup E_P)$ is eulerian (i.e. connected with all nodes having even degree).

If $G_{R\cup P}$ is eulerian, it has an eulerian cycle that traverses each edge in $E_R\cup E_P$ exactly once. Note that since E_P is a multiset, it may contain more than one copy of an edge *e* in E_D.

The above definition is equivalent to the first definition given above since the costs d_{ij} correspond to the cost of a minimum cost path between *i* and *j* in the original graph $G=(V, E)$ and E_R is the set of required edges.

Several lemmas are proposed in [5] that lead to a structural characterization of a subset of optimal solutions for the RPP. These lemmas are also the basis for a new integer programming formulation of the RPP, called the RPP(GW) model. The main advantages of this new formulation are: First, the number of constraints is a linear function on the number of vertices of the graph. Second, the set of feasible solutions to this formulation, which includes some optimal solutions, is a subset of all tours to the RPP. And third, the problem is now reduced to find a minimum cost perfect matching in a given graph as will be seen below.

Let $E_D=\{\{i,j\}/i\in V_R$ and $j\in V_R$, $i\neq j\}$, with costs d_{ij} associated to each $\{i,j\}$. Denote $C(i)$ the connected component of G_R that contains vertex i. The problem of finding a tour for the RPP, or equivalently, a multiset E_P of edges in E_D with minimum total cost, such that $(V, E_P\cup E_R)$ is eulerian, can then be formulated as follows:

First, build a new graph $G_A=(V_A, E_A)$ from G_R. V_A contains all nodes in V_R plus a new node i' for each R-even vertex i. E_A contains edges connecting every pair i,j of nodes in V_A, except edges $\{i,j\}$, $\{i',j\}$ y $\{i',j'\}$, when j is in the same component $C(i)$ as i, $d_R(i)$ is even and $j\neq i$. The cost of an edge $\{i,j\}$ is d_{ij}, the cost of $\{i,i'\}$ is zero, and $d_{ij'} = d_{j'i} = d_{ij}$.

Let $E_A^* = \{\{k,l\}\in E_A, C(R(k))\neq C(R(l))\}$ where $R(k)=i$ if $k=i$ or $k=i'$. Let E_k^+ be the set of edges incident on node k in G_A.

Let M_A be a perfect matching in G_A. Let E_p be the multiset of edges obtained as follows: if $\{i,j\}\in M_A$ with $R(i)\neq R(j)$, then $\{R(i),R(j)\}\in E_p$. As can be seen, finding an optimal solution for the RPP is equivalent to finding a minimal cost perfect matching in G_A such that (V, $E_P \cup E_R$) is connected, where E_P is obtained from the optimal perfect matching.

RPP(GR): Variant of RPP(GW):

We define variables $\boldsymbol{x_{ij}}$, $i \leq j$, associated to each edge {i,j} in E_A.

Let C_1, C_2, ... C_m be the connected components of G_R. For each pair of components C_l, C_k, k=1,...,m and l=2,...,m, define a flow variable $\boldsymbol{y_{kl}}$. This new formulation reduces both, the number of flow variables and the number of "link" constraints with respect to RPP(GW). RPP(GW) has a flow variable for each edge in E_A^*.

The formulation is as follows:

$$(1) \qquad \min \sum_{\{i,j\} \in E_A} d_{ij} x_{ij}$$

subject to:

$$(2) \qquad \sum_{\{i,j\} \in E_k^+} x_{ij} = 1, k \in V_A$$

$$(3) \qquad \sum_{k \in \{2,\ldots,m\}} y_{1k} = m-1$$

$$(4) \qquad \sum_{k \neq t} y_{kt} - \sum_{k \neq t} y_{tk} = 1 \quad t = 2,\ldots,m$$

$$(5) \qquad y_{kl} \leq (m-1) \times \sum_{\substack{\{i,j\} \in E_A^*: \\ (i \in C_k \wedge j \in C_l) \vee \\ (j \in C_k \wedge i \in C_l)}} x_{ij} \ , \quad k \neq l, \quad k = 1,\ldots,m \qquad l = 2,\ldots,m$$

$$(6) \qquad x_{ij} \in \{0,1\}, \{i,j\} \in E_A$$

$$(7) \qquad y_{kl} \geq 0, \quad k \neq l, \quad k = 1,\ldots,m \qquad l = 2,\ldots,m$$

As it has been mentioned, finding an optimal solution for the RPP is equivalent to finding a minimal cost perfect matching in G_A such that (V,$E_P \cup E_R$) is connected, where E_P is obtained from the optimal matching. Constraints (1), (2) y (6) define a minimum cost perfect matching in G_A. Constraints (3),(4) and (7) define a network flow problem on the *y* variables. This flow induces a spanning connected subgraph on the reduced graph of G_A (whose vertices are the connected components of G_A), in which *m* units of flow must be sent from component C_1 to the other components. These variables are used to guarantee connectivity which is established in (5), called the "link" constraints, relating both types of variables.

3. Some valid inequalities for RPP(GR)

A valid inequality is an inequality satisfied by all feasible solutions. In this section we present some valid inequalities for RPP(GR).

(1) Connectivity Inequalities (Corberán et al. [2]):

Since (V_R, $E_R \cup E_P$) has an Eulerian cycle, for $J \subset \{1, \ldots, m\}$ the edge-cut set $\Omega(\cup C_j : j \in J)$ has an even number of edges:

$$\sum_{\substack{R(k)\in C_j,\, j\in J \\ R(l)\notin \bigcup_{i\in J} C_i}} x_{kl} \geq 2,$$

(2) Matching inequalities (Jack Edmonds [10]):

$$\sum_{\substack{R(k),R(l)\in S, \\ R(k)\neq R(l)}} x_{kl} \leq \frac{|S|-1}{2}, \quad S \subset V_R,\ |S| \geq 3,\ |S| \text{ odd}$$

(3) The K-C inequalities (Corberán and Sanchis [1]) can be adapted to RPP(GR):

A K-C configuration is a partition $\{V_0, \ldots, V_K\}$ of V_R, with $K \geq 3$, such that :

- $V_1, \ldots, V_{K-1}$ and $V_0 \cup V_K$ are cluster of one or more R-sets,
- $|E_R(V_0:V_K)|$ is positive and even,
- $|E_A(V_i^A : V_{i+1}^A)| \neq \varnothing$ for i=0, … , K-1. Where $E_A(S:T)$ is the set of edges of E_A with one end vertex in S and the other end vertex in T, V_i^A is the union of V_i and $\{i' : i \in V_i$ with $d_R(i)$ even $\}$

The corresponding K-C inequality to RPP(GR) can be written as:

$$(K-2)x(E_A(V_0^A : V_K^A)) + \sum_{\substack{0\leq i<i\leq K \\ (i,j)\neq(0,K)}} |i-j|\, x(E_A(V_i^A : V_j^A)) \geq 2(K-1)$$

4. A Cutting Plane algorithm for RPP

In this section we present a cutting plane algorithm (the MCK algorithm) in order to find lower bounds for RPP. The algorithm was tested with the same problem set as in [2].

In [2], Corberán et al. present a cutting plane algorithm for RPP(COR) using non-negativity, connectivity, R-odd, K-Component, Path-bridge and Honey Comb inequalities. Although this worked adequately for the instances tested, the algorithm added a large number of cutting planes in general. The computational study on our algorithm shows that –on the average- very few cutting planes suffice to find a globally optimal solution and it suffices to consider connectivity, matching and K-Component inequalities only. We took as the initial LP relaxation of our algorithm the linear relaxation of RPP(GR) in order to reduce the number of flow variables.

4.1. The Matching-Connectivity-K-C Cutting Plane Algorithm (or MCK algorithm)

(1) Let LP be the linear program having all the inequalities of RPP(GR) and all integer variables relaxed to real numbers. Go to Step (2).
(2) Let x* be the optimal solution of LP. Go to Step (3)
(3) If x* is integer (all its components are integer numbers) then go to step (9). Otherwise, go to Step (4).
(4) Look for some Matching and Connectivity inequalities violated by x* (if there exist violated inequalities, this step finds at least one them). Go to Step (5)
(5) If some violated inequalities have been found in step (4) then add all of them to LP and go to Step (2). Otherwise go to Step (6)
(6) Look for some KC inequalities violated by x* (this step is a heuristic algorithm, so it is possible that there exist KC inequalities violated by x* but this step does not find any). Go to Step (7)
(7) If a violated inequality has been found in Step (6) then add all such found inequalities to LP and go to Step (2). Otherwise, go to Step (8)
(8) If the objective function value VAL for x* is not integer then add to LP the inequality: objective function greater or equal to the nearest integer greater than VAL, and assign to x* the optimal sotution of LP. Go to Step (9)
(9) x* is the solution found by the MCK algoritm, STOP. (If x* is integer then we have obtained an optimal solution for the RPP problem whose model is RPP(GR))

Remarks:

(a) The number of iterations of MCK is the number of times that step (2) is executed, that is the number of linear programs that are solved.

(b) Explanation of Step (4):

Let G_A be the graph from which the RPP(GR) model has been formulated. Let C_i be the set of vertices of G_A corresponding with the connected component *i* of G_R.

Given a proper vertex subset X of G_A, the cut–edge set of G_A corresponding with X will be denoted by $\Omega_A(X)$. We say that $\Omega_A(X)$ is an odd cut if the cardinality of X is odd. For a vertex set X, E(X) denotes the set of edges of G_A having its two end vertices in X.

A matching inequality is: $$\sum_{e \in E(X)} x(e) \le (|X|-1)/2$$

where *X* is a proper subset of vertices of G_A with odd cardinality and *x(e)* is the variable associated to edge *e* in RPP(GR).

The above inequality is equivalent to the following inequality: $\sum_{e \in \Omega_A(X)} x(e) \geq 1$ (see [7]).

A connectivity inequality is: $\sum_{e \in \Omega_A(U)} x(e) \geq 2$

where $\Omega_A(U)$ is a cut set and U is the union of some C_i's.

We know that matching and connectivity inequalities are valid for RPP(GR).

There are two procedures for finding violated inequalities: one looks for violated Matching inequalities and the other one looks for violated Connectivity inequalities. The two procedures are exact (not heuristics), in the sense that if there is a violated Matching or Connectivity inequality then the respective procedure finds it.

As we will see later, more than one violated Matching and/or Connectivity inequality can be found at each iteration. Let x* be the current optimal solution of LP. We say that the weight of edge *e* is *x*(e).* G_A(x*) is the graph with the same vertex set as G_A and set of edges G_A(x*) with those edges of G_A having x*(e) not zero.

<u>Separation procedure for the Matching Inequalities:</u>

Let V_1 denote the set of vertices of G_A(x*) that are incident to an edge of G_A(x*) with weight one. Remove, from G_A(x*), V_1 and all edges incident to a vertex in V_1. Denote the new graph by G'(x*). Clearly each minimum weighted odd cut in G'(x*) determines a minimum weighted odd cut in G_A(x*), and viceversa. (Note also that the number of vertices of G'(x*) is even). So, x* violates a Matching inequality if and only if there is an odd cut with weight less than one in the graph G'(x*).

The minimum cut values between each of the *n(n-1)/2* pairs of vertices of a order *n* graph G can be represented by a weighted tree *T* on *n* vertices called a Gomory-Hu tree, where for any pair of vertices *(x,y)*, if *e* is the minimum weigt edge on the path from *x* to *y* in *T*, then the minimum cut value between *x* and *y* in G is exactly the weigt of *e*. Moreover, the two components of *T-e* form a minimum cut between *x* and *y* in G.

As shown by Padberg and Rao in [9] a minimum weighted odd cut of G'(x*) can be found by determining the Gomory-Hu tree of G'(x*) and finding an edge of this tree having minimum capacity among those edges whose removal splits the tree into two components of odd size.

Our procedure constructs a Gomory-Hu tree T. Then, for each edge *e* of T with weight less than one that defines an odd cut, we take the set of vertices X of the component of *T-e* with minimum cardinality, and generate the following violated Matching inequality:

$\sum_{e \in E(X)} x(e) \leq (|X|-1)/2$ where *x(e)* is the variable corresponding with *e* in RPP(GR)

Hence, it is possible to generate up to *(n-1)* violated Matching inequalities, where *n* is the order of G'(x*).

<u>Separation procedure for the Connectivity Inequalities:</u>

Let G''(x*) be the graph obtained from G_A(x*) by shrinking (or reducing) each Ci, $1 \leq i \leq m$, into a vertex v_i and connecting vertex v_i with v_j by an edge with weight equal to the sum of the weights of edges in G_A(x*) with an extremity in C_i and the other extremity in C_j.

Our procedure constructs a Gomory-Hu tree T on G''(x*) and for each edge *e* of *T* with weight less than two we take the set of vertices X of the component of *T-e* with minimum cardinality, and generate the following violated Connectivity inequality:

$$\sum_{e \in \Omega_A(U)} x(e) \geq 2$$

where $\Omega_A(U)$ is the cut set of G_A associated to *U* (the union of the C_i for v_i in X).

Hence, it is possible to generate up to *(n-1)* violated Connectivity inequalities, where *n* is the order of G'(x*).

(c) The heuristic algorithm looking for violated K-C inequalities in step (6) is similar to the one of Corberan el al. (see [2]). The heuristic of Corberán et al. has been adapted to cope with graph G_A where even degree vertices in G_R are duplicated in G_A (see Section 3).

In [2], given a R-set C_i, a vertex in C_i is called *x*-external* if it is connected to at least one vertex in a different R-set by an edge *e* with $x_{*(e)} > 0$. In the first phase, Corberán et al. [2] look for " …an x*-external vertex *u* whose corresponding connected component in the subgraph has an even number of R-odd vertices…". Instead, we first look for a x*-external vertex *u* whose corresponding connected component in the subgraph has two vertices (i.e., it corresponds with a R-even vertex of G_R). If such a component does not exist, we look for components C in the subgraph with |V(C)|-[x*(Ω(C))] odd (i.e., with an odd number of vertices not matched. [x] means the greater integer less than x). Finally if there is not such a component we apply the criterion of Corberán et al. "… a component with an even number of R-odd vertices…". In the second phase Corberán et al. " …compute a spanning tree by iteratively adding the edge of maximum weight not forming a cycle… ". We compute up to three such trees by modifying the criterion of adding edges in order to find a violated K-C inequality. The first tree is like in [2], the second tree is obtained by reversing the order of adding edges in [2], and finally, the third tree is the one with the longest path leaving from V_0.

5. Computational Results

The MCK cutting plane algorithm has been coded in the C Language and run on a SUN Sparc Station 10/30 with 4 HyperSparc at 100 MHz using only one processor. The CPLEX 4.0 library has been used to solve LP problems. The algorithm has been tested on the same problem set than in [2] corresponding to 118 RPP problems.

Tables 1 to 4 correspond to the MCK cutting plane results. The column headings are defined as follows:

(a) Algorithm Value (MCK): Solution value of the MCK cutting plane algorithm. An "*" on an entry means that it was necessary to append an extra inequality to enforce integrality of the objective value.
(b) Optimal Value (OV): Optimal solution value of the RPP instance.
(c) MCK/OV: Solution value of the MCK cutting plane algorithm divided by the optimal solution value.
(d) Time in Secs.: CPU time in seconds.

	Algorithm Value (MCK)	Optimal Value (OV)	MCK/ OV	Time in Secs.
ALBAIDAA	3304	3304	1	64.93
ALBAIDAB	2826	2826	1	42.31
P01	51	51	1	0.06
P02	72	72	1	0.15
P03	29	29	1	0.43
P04	29	29	1	0.16
P05	55	55	1	0.41
P06	32	32	1	0.3
P07	37	37	1	0.12
P08	29	29	1	0.08
P09	26	26	1	0.1
P10	35	35	1	0.15
P11	9	9	1	0.05
P12	6	6	1	0.04
P13	23	23	1	0.04
P14	57	57	1	2.21
P15	261	261	1	0.85
P16	64	64	1	5.92
P17	49	49	1	0.65
P18	85	85	1	0.39
P19	116	116	1	0.81
P20	116	116	1	23.13
P21	78*	78	1	84.32
P22	122	122	1	5.92
P23	95	95	1	22.99
P24	113*	113	1	6.06

Table 1: Computational results of the MCK algorithm for the Christofides&al (1981) and Corberán&Sanchis (1994) instances.

RANDOMPR	Algorithm Value (MCK)	Optimal Value (OV)	MCK/ OV	Time in Secs.
0	20716	20716	1	0.04
1	23136	23136	1	0.04
2	14677	14677	1	0.03
3	16576	16576	1	0.05
4	7552	7552	1	0.04
5	20459	20459	1	0.05
6	16902	16902	1	0.05
7	20729	20729	1	0.07
8	23974	23974	1	0.08
9	19425	19425	1	0.2
10	16850	16850	1	0.27
11	27413	27413	1	0.19
12	39240	39240	1	0.09
13	19707	19707	1	0.63
14	23391	23391	1	0.52
15	23193	23193	1	1.06
16	30960	30960	1	0.53
17	27311	27311	1	0.13
18	21209	21209	1	0.31
19	13640	13640	1	0.71

Table 2: Computational results of the MCK algorithm for Class 1 RPP random instances from Hertz&al. (1998).

GRID PROB	Algorithm Value (MCK)	Optimal Value (OV)	MCK/ OV	Time in Secs.
0	9	9	1	0.04
1	9	9	1	0.08
2	8	8	1	0.06
3	8	8	1	0.08
4	7	7	1	0.09
5	5	5	1	0.07
6	7	7	1	0.43
7	6	6	1	0.08
8	7	7	1	0.1
9	11	11	1	0.26
10	15	15	1	0.61
11	15	15	1	0.74
12	14	14	1	1.13
13	17	17	1	1.09
14	17*	17	1	6.69
15	13	13	1	0.68
16	16	16	1	2.52
17	14	14	1	0.6
18	22	22	1	1.2
19	23	23	1	13.22
20	25	25	1	6.7
21	22	22	1	24.59
22	19	19	1	2.8
23	24	24	1	5.01
24	22	22	1	4.61
25	21	21	1	2.41
26	16	16	1	2.86
27	37	37	1	73.12
28	35	35	1	31.79
29	38	38	1	25.19
30	37	37	1	31.74
31	37	37	1	78.28
32	37*	38	0.9736	64.51
33	35	35	1	6.69
34	37*	37	1	32.17
35	33	33	1	20.54

Table 3: Computational results of the MCK algorithm for Class 2 RPP random instances from Hertz&al. (1998)

DEGREE PROB	Algorithm Value (MCK)	Optimal Value (OV)	MCK/OV	Time in Secs.
0	153	153	1	0.02
1	451	451	1	0.06
2	380	380	1	0.1
3	411	411	1	0.04
4	521	521	1	0.08
5	473	473	1	0.22
6	504	504	1	0.21
7	457	457	1	0.08
8	399	399	1	0.36
9	666	666	1	0.44
10	488	488	1	0.27
11	459	459	1	0.84
12	571	571	1	0.56
13	347	347	1	0.72
14	458	458	1	0.83
15	492	492	1	0.74
16	354	354	1	2
17	448	448	1	1.25
18	547	547	1	1.7
19	644	644	1	2.87
20	569	569	1	1.08
21	451	451	1	4.59
22	589	607	0.97035	4.68
23	600	600	1	3.31
24	465	465	1	2.46
25	515	515	1	8.96
26	573	573	1	1.64
27	870*	872	0.99771	143.17
28	780	780	1	16.6
29	841*	846	0.99409	61.35
30	722	722	1	81.91
31	739	739	1	9.7
32	659	659	1	118.77
33	746*	752	0.9920	405.78
34	683	683	1	13.51
35	638	638	1	34.23

Table 4: Computational results of the MCK algorithm for Class 3 RPP random instances from Hertz&al. (1998).

In Tables 1 to 4 it can be seen that 113 out of 118 RPP instances were solve to optimality and among the other 5, 2 had a MCK/OV ratio of 97% and the other 3 a ratio of 99%. The average number of identified violated inequalities in the 118 problems is 31. In Corberán et al. (see [2]) this average was 93, twice as much as ours. Thus, for these instances the algorithm works extremely well.

6. Conclusions

We have formulated the model RPP(GR), variant of RPP(GW). The number of y variables, corresponding to flow or tree edges used to achieve connectivity of the solution, are considerably reduced in this formulation.

Our goal in this research work has been to explore the possibility of finding good lower bounds for the RPP using Garfinkel and Webb's variant RPP(GR) .

The computational results given in Section 5 show that our MCK cutting plane algorithm is capable of solving almost all instances to proven optimality. There is still room for improvement in our algorithms, specially reducing CPU time. One improvement would be the addition of heuristic separation algorithms for Matching and Connectivity inequalities in order to decrease the running time. Another improvement would be the reduction of variables when solving the LP problems in each iteration. This can be accomplished by adopting a scheme of column addition as presented in [7].

Acknowledgements: This research was partially funded by the Spanish Agency of Iberoamerican Cooperation (A.E.C.I.). Part of this work was carried out while the second author was on sabbatical leave at Polytechnique University of Catalunya, Spain. This support is gratefully acknowledged. The authors want to thank Angel Corberán and Alain Hertz for providing us their instance sets.

7. References

[1] A. Corberán, J.M. Sanchis. “A Polyhedral approach to the Rural Postman Problem”. European Journal of Op. Res. vol. 79, 95-114. 1994.

[2] A. Corberán, A. Letchford, J. M. Sanchis. “A Cutting Plane Algorithm for the General Routing Problem”. Personal communication. Nov. 1998.

[3] N. Christofides, V. Campos, A. Corberán, E. Mota. "An Algorithm for the Rural Postman problem". Imperial College Report IC.O.R.81.5. 1981.

[4] G.N. Frederickson. “Approximation algorithms for some Postman problems”. Journal for the Association for Computing Machinery, 7, 178-193. 1979

[5] R. Garfinkel, I. Webb. “On Crossings, the Crossing Postman Problem, and the Rural Postman Problem”. To appear in Networks.

[6] G. Ghiani, G. Laporte. “ A branch-and-cut algorithm for the undirected rural postman problem”. Center for research on Transportation. Universidad de Montréal. Report CRT-97-54. Dec. 1997.

[7] M. Grötschel, O. Holland. “Solving Matching Problems with Linear Programming”. Mathematical Programming, vol. 33, 243-259, 1985

[8] A. Hertz, G. Laporte, P. Nanchen. “Improvement procedures for the undirected rural postman problem”. Working paper. Center for Research on Transportation, Univ. Montréal. 1996

[9] M.W. Padberg, M.R. Rao, “Odd Minimum Cut-Sets and b-Matchings”. Mathematics of Operations Research 7, 67-80, 1982.

[10] Jack Edmonds. “Maximum Matching and a polyhedron with 0, 1 vertices”. Journal of Research National Bureau of Standrads 69B (1965) 125-130.