Math. Program., Ser. A (2013) 141:21–48
DOI 10.1007/s10107-011-0507-6

FULL LENGTH PAPER

A branch-and-cut algorithm for the maximum benefit Chinese postman problem

Ángel Corberán · Isaac Plana · Antonio M. Rodríguez-Chía · José M. Sanchis

Received: 16 March 2011 / Accepted: 6 December 2011 / Published online: 16 December 2011

Abstract The Maximum Benefit Chinese Postman Problem (MBCPP) is an NP-hard problem that considers several benefits associated with each edge, one for each time the edge is traversed with a service. The objective is to find a closed walk with maximum benefit. We propose an IP formulation for the undirected MBCPP and, based on the description of its associated polyhedron, we propose a branch-and-cut algorithm and present computational results on instances with up to 1,000 vertices and 3,000 edges.

Keywords Chinese postman problem · Maximum benefit Chinese postman problem · Rural postman problem · Facets · Branch-and-cut

Mathematics Subject Classification (2000) 90C57 · 90C27

1 Introduction

The purpose of this paper is to propose a branch-and-cut algorithm for the undirected Maximum Benefit Chinese Postman Problem (MBCPP) defined as follows. Let $G = (V, E)$ be an undirected connected graph where V is the vertex set and E is the edge set. Vertex $1 \in V$ represents the depot. Each edge $e \in E$ has $n_e \geq 0$ benefits,

Á. Corberán (✉)
Dept. de Estadística e Investigación Operativa, Universitat de València, Valencia, Spain
e-mail: angel.corberan@uv.es

I. Plana
Dept. Matemáticas para la Economía y la Empresa, Universitat de València, Valencia, Spain

A. M. Rodríguez-Chía
Dept. de Estadística e Investigación Operativa, Universidad de Cádiz, Cádiz, Spain

J. M. Sanchis
Dept. de Matemática Aplicada, Universidad Politécnica de Valencia, Valencia, Spain

$b_e^1, \ldots, b_e^{n_e}$, giving the gross benefit of servicing the edge for the first, second,..., n_eth time. Moreover, each edge $e \in E$ has $n_e + 1$ different associated costs, $c_e^1, \ldots, c_e^{n_e}$ and c_e^d. The n_e first ones represent the cost of traversing and servicing edge e for the first, second,..., n_e-th time, while the last one corresponds to the cost of just traversing that edge without servicing it (the deadhead cost). Therefore, the net benefit of the tth traversal of edge e is given by $b_e^t - c_e^t$ for $t = 1, \ldots, n_e$, while the net benefit of deadheading an edge is $-c_e^d$. The MBCPP then consists of finding a tour starting from the depot, traversing some of the edges in E a certain number of times and returning to the depot, with maximum total net benefit. The MBCPP is NP-hard because the Rural Postman Problem [18], which was proved to be NP-hard in [14], can be considered a special case of the MBCPP [20].

The MBCPP was introduced by Malandraki and Daskin [16], who studied its directed version. They modeled it as a minimum cost flow problem with subtour elimination constraints. Based on this approach, they proposed a branch-and-bound procedure and solved instances on a network of 25 vertices.

Applications of this problem arise in the routing of street cleaners and the design of street snow-plowing and snow-salting routes. These applications are usually formulated as traditional arc routing problems in which only one single service traversal on each edge is considered. However, an additional benefit can be derived when a street is plowed multiple times and the benefit may depend upon whether the link represents a main road or a low-density-traffic street. As is pointed out in [16], *"the key advantage of this approach over the traditional approach is that it allows links to be excluded from the solution or included multiple times if doing so is advantageous"*.

The scientific literature on the MBCPP is rather limited. In [20], an approximate algorithm for solving the MBCPP on undirected graphs is devised. The procedure is illustrated on an example with 15 vertices and 26 edges. The algorithm expands the original graph by replacing each edge with a set of edges of positive net benefit. Minimal spanning tree and matching algorithms are then applied to generate a postman tour. In [19], several heuristic algorithms for solving the MBCPP on directed graphs are proposed. The authors report computational results on graphs with up to 30 vertices and 780 arcs. In all these papers it is assumed that $c_e^1 = \cdots = c_e^{n_e} \geq c_e^d$ and $b_e^1 \geq \cdots \geq b_e^{n_e} \geq 0$.

Some related problems have also been studied. Among them, the Prize-Collecting Arc Routing Problem (PCARP), also called Privatized Rural Postman Problem. In the PCARP only the edges in a given subset of edges $D \subseteq E$ have an associated benefit and it is assumed that this benefit can be collected only once, independently of the number of times the edge is traversed. Note that this problem is a special case of the MBCPP in which $n_e = 1$ for all the edges in D, while $n_e = 0$ for the remaining ones. The PCARP was introduced in [3], where an ILP formulation with binary variables was provided. In particular, the authors used a new family of inequalities, called set-parity inequalities, which are an adaptation to this problem of the so-called cocircuit inequalities introduced in [5]. In [2], an LP-based algorithm for solving the problem on undirected graphs was proposed. A related problem, the Clustered Prize-collecting Arc Routing Problem, has recently been studied in [1] and [12] for undirected graphs and in [6] for 'windy' graphs. Other arc routing problems with benefits have been studied in [11] and [4].

The scientific contributions of this paper are: a more general and useful framework for the undirected MBCPP, a formulation for it, a polyhedral study, and a branch-and-cut algorithm for its exact resolution.

The remainder of this paper is organized as follows. In Sect. 2 some notation for the problem is introduced and an IP formulation is presented. The polyhedron associated with the MBCPP is defined, and its dimension and some basic facet-inducing inequalities are studied in Sect. 3. Section 4 then introduces three non-trivial families of valid inequalities, namely parity, K-C and p-connectivity inequalities. The separation problems associated with all the families of valid inequalities presented are discussed in Sect. 5, as well as the features of the proposed branch-and-cut algorithm. The computational results obtained with this procedure on different sets of instances are presented in Sect. 6. Finally, some concluding comments are given in Sect. 7.

2 Problem formulation

Consider an undirected and connected graph $G = (V, E)$. Associated with each edge $e \in E$, there are $n_e + 1$ net benefits. The n_e first ones, $\overline{b}_e^t = b_e^t - c_e^t$, $t = 1, \ldots, n_e$, correspond to the traversals of the edge servicing it, while the last one, $\overline{b}_e^{n_e+1} = -c_e^d$, is associated with the deadheading of e. Note that additional traversals of this edge would have an associated net benefit of $-c_e^d$.

We first prove that the above instance can be transformed into an equivalent one with only two net benefits associated with each edge.

Theorem 1 *Solving the MBCPP on a graph with $n_e + 1$ net benefits associated with each edge e is equivalent to solving it with only two net benefits, b_e^{odd} and b_e^{even}, for the first and the second traversals of each edge e, respectively, where*

$$b_e^{odd} = \max\left\{\sum_{\ell=1}^{k} \overline{b}_e^{\ell} \ : \ k \text{ is odd and } k \leq n_e + 1\right\}$$

$$b_e^{even} = \max\left\{\sum_{\ell=1}^{k} \overline{b}_e^{\ell} \ : \ k \text{ is even and } k \leq n_e + 1\right\} - b_e^{odd}.$$

If $n_e = 0$, we define $b_e^{even} = b_e^{odd} = -c_e^d$.

Proof Consider an edge e with net benefits $\overline{b}_e^1, \ldots, \overline{b}_e^{n_e+1}$. If edge e is traversed an odd number of times in an optimal solution, then it will be traversed exactly $k^{odd} = \text{argmax}_k \left\{\sum_{\ell=1}^{k} \overline{b}_e^{\ell} \ : \ k \text{ is odd and } k \leq n_e + 1\right\}$ times to get the maximum benefit from servicing this edge, which will be $\sum_{\ell=1}^{k^{odd}} \overline{b}_e^{\ell}$. This is equivalent to traversing and servicing this edge exactly once with net benefit b_e^{odd}. If e is traversed an even number of times, it will be traversed $k^{even} = \text{argmax}_k \left\{\sum_{\ell=1}^{k} \overline{b}_e^{\ell} \ : \ k \text{ is even and } k \leq n_e + 1\right\}$ times with a total net benefit of $\sum_{\ell=1}^{k^{even}} \overline{b}_e^{\ell}$. In this case, this is equivalent to traversing the edge twice, the first time with net benefit b_e^{odd} and the second one with b_e^{even}. □

Consider for instance an edge with the following net benefits: 4, 2, −1, 2, −4.If this edge is traversed an odd number of times in an optimal solution, then it will be traversed exactly 3 times, since in this way a maximum net benefit of 5 units is obtained. Similarly, if it is traversed an even number of times in an optimal solution, this number should be 4 and the net benefit obtained would be 7. Therefore, we can solve the same problem in a graph where this edge has only two net benefits $b^{odd} = 5$ and $b^{even} = 2$. Note that if this edge is traversed once a net benefit of 5 is obtained, while if it is traversed twice, we obtain a net benefit of 5+2 units.

As a consequence of the above theorem, the MBCPP can be formulated using two binary variables x_e and y_e for each edge $e = (i, j) \in E$. Variable x_e takes the value 1 if e is traversed and 0 if e is not traversed, while variable y_e takes the value 1 if e is traversed twice and 0 otherwise. In other words, variables x_e and y_e represent the first and second traversal of edge e, respectively.

In this paper we use the following notation. Given two subsets of vertices $S, S' \subseteq V$, $(S : S')$ denotes the edge set with one endpoint in S and the other in S'. Given a subset $S \subseteq V$, let us denote $\delta(S) = (S : V \backslash S)$ and let $E(S) = \{e = (i, j) \in E : i, j \in S\}$ be the set of edges with both endpoints in S. Finally, for any subset $F \subseteq E$, $x(F)$ denotes $\sum_{e\in F} x_e$ and $y(F) = \sum_{e\in F} y_e$, while $(x - y)(F) = \sum_{e\in F} (x_e - y_e)$ and $(x + y)(F) = \sum_{e\in F} (x_e + y_e)$.

We propose the following formulation for the MBCPP:

$$\text{Maximize} \qquad \sum_{e\in E} \left(b_e^{odd} x_e + b_e^{even} y_e\right)$$

s.t.:

$$\sum_{e\in\delta(i)} \left(x_e + y_e\right) \equiv 0 \pmod 2, \quad \forall i \in V \tag{1}$$

$$\sum_{e\in\delta(S)} \left(x_e + y_e\right) \geq 2x_f, \quad \forall S \subset V\backslash\{1\}, \quad \forall f \in E(S) \tag{2}$$

$$x_e \geq y_e, \quad \forall e \in E \tag{3}$$

$$x_e, y_e \in \{0, 1\}, \quad \forall e \in E. \tag{4}$$

Constraints (1) force the solution to visit each vertex an even number of times, possibly 0. Conditions (2) ensure the connectivity of the solution, and constraints (3) guarantee that a second traversal of an edge can only occur when it has been traversed previously. Note that constraints (1) are not linear, although they could be linearized by introducing additional integer variables, which would give an Integer Linear Programming formulation. Note also that $(x, y) = (0, 0)$ satisfies the above constraints and is, therefore, a feasible solution to the MBCPP.

3 MBCPP polyhedron

In this section we study the polyhedron associated with the MBCPP. In particular, its dimension is obtained and some basic inequalities are proved to be facet-inducing.

Let us call each vector $(x, y) \in \{0, 1\}^{2|E|}$ satisfying (1) to (4) an MBCPP tour and let MBCPP(G) be the convex hull of all MBCPP tours. Obviously, it is a polytope.

Remember that a graph G is called 3-edge connected if every proper cut-set $\delta(S)$, $S \subset V$, contains at least 3 edges. It is well known that G is 3-edge connected if, and only if, for every pair of vertices $i, j \in V$, there are at least three edge-disjoint paths in G connecting i and j.

Theorem 2 *MBCPP(G) is a full-dimensional polyhedron (dim(MBCPP(G))=* $2|E|$*) if, and only if, G is 3-edge connected.*

Proof If G is not 3-edge connected there is a cut-set $\delta(S)$ with at most 2 edges. If $\delta(S)$ contains exactly two edges, namely e and f, it can be seen that all MBCPP tours satisfy the equation $x_e - y_e = x_f - y_f$. Moreover, if $\delta(S) = \{e\}$, then all MBCPP tours satisfy $x_e = y_e$. Therefore, in both cases, the polyhedron is not full-dimensional.

On the other hand, let us now suppose that graph G is 3-edge connected. We will prove that the polyhedron is full-dimensional. Let $ax + by = c$ (that is, $\sum_{e\in E} a_e x_e + \sum_{e\in E} b_e y_e = c$) be an equation satisfied by all the MBCPP tours. We have to prove that $a = b = c = 0$.

Given that $(x, y) = (0, 0) \in$ MBCPP(G), $a \cdot 0 + b \cdot 0 = c$ holds and then $c = 0$. Let $(i, j) \in E$ be an arbitrary edge. Since G is connected, there is a path $\mathcal{P}$ joining vertices 1 and i (or j) not containing edge (i, j). The solution that traverses the path $\mathcal{P}$ twice (that is, $x_e = y_e = 1 \ \forall e \in \mathcal{P}$) is an MBCPP tour and then $\sum_{e\in\mathcal{P}} a_e + \sum_{e\in\mathcal{P}} b_e = 0$ holds. On the other hand, the solution that traverses the path $\mathcal{P}$ and the edge (i, j) twice is also an MBCPP tour and then $\sum_{e\in\mathcal{P}} a_e + \sum_{e\in\mathcal{P}} b_e + a_{ij} + b_{ij} = 0$. By subtracting both expressions, we obtain $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E$.

Let $\mathcal{C}$ be any cycle in graph G. There is a path $\mathcal{P}$ joining vertex 1 and a vertex i in the cycle. The solution that traverses the path $\mathcal{P}$ twice ($x_e = y_e = 1$) and the cycle $\mathcal{C}$ once ($x_e = 1, y_e = 0$) is an MBCPP tour and then $\sum_{e\in\mathcal{P}} a_e + \sum_{e\in\mathcal{P}} b_e + \sum_{e\in\mathcal{C}} a_e = 0$. Given that $a_{ij} + b_{ij} = 0$, we obtain $\sum_{e\in\mathcal{C}} a_e = 0$ for any cycle $\mathcal{C}$ in G.

Let $(i, j) \in E$ be an arbitrary edge. Since G is 3-edge connected, there are two edge-disjoint paths $\mathcal{P}_1, \mathcal{P}_2$ joining vertices i and j that do not contain edge (i, j). Then, considering the three cycles $\mathcal{P}_1 \cup \{(i, j)\}$, $\mathcal{P}_2 \cup \{(i, j)\}$, and $\mathcal{P}_1 \cup \mathcal{P}_2$, for which $\sum_{e\in\mathcal{C}} a_e = 0$, we obtain $a_{ij} = 0$. Given that $a_{ij} + b_{ij} = 0$, we obtain $b_{ij} = 0$ for each edge $(i, j) \in E$. Hence, $a = b = c = 0$ and the polyhedron MBCPP(G) is full-dimensional. □

In the following, we will assume that graph G is 3-edge connected and thus MBCPP(G) is full-dimensional. Therefore, every facet of the polyhedron is induced by a unique inequality (except scalar multiples).

Theorem 3 *Inequality* $y_{uv} \geq 0$*, for each edge* $(u, v) \in E$*, is facet-inducing of MBCPP(G) (if graph G is 3-edge connected).*

Proof Let us suppose there is another valid inequality $ax + by \geq c$ such that

$$\{(x, y) \in \text{MBCPP}(G) : \ \ y_{uv} = 0\} \subseteq \{(x, y) \in \text{MBCPP}(G) : \ \ ax + by = c\}.$$

We will prove that inequality $ax + by \geq c$ is a scalar multiple of $y_{uv} \geq 0$. Given that the tour $(0, 0)$ satisfies $y_{uv} = 0$, it follows that $c = 0$. A similar argument to that used in the proof of Theorem 2 leads to $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E\backslash\{(u, v)\}$, and $\sum_{e\in\mathcal{C}} a_e = 0$ for any cycle $\mathcal{C}$ in G. Then, as above, we obtain $a_{ij} = 0$ for each edge $(i, j) \in E$. Since $a_{ij} + b_{ij} = 0$, also $b_{ij} = 0$ for each edge $(i, j) \in E\backslash\{(u, v)\}$. Thus, inequality $ax + by \geq c$ turns out to be $b_{uv} y_{uv} \geq 0$ and $y_{uv} \geq 0$ is facet-inducing for MBCPP(G). □

Theorem 4 *Inequality $x_{uv} \leq 1$, for each edge $(u, v) \in E$, is facet-inducing for MBCPP(G) (if graph G is 3-edge connected).*

Proof Let us suppose there is another valid inequality $ax + by \leq c$ such that

$$\{(x, y) \in \text{MBCPP}(G) : \ \ x_{uv} = 1\} \subseteq \{(x, y) \in \text{MBCPP}(G) : \ \ ax + by = c\}.$$

Let $(i, j) \in E\backslash\{(u, v)\}$. Given that G is 3-edge connected, graph $G\backslash\{(i, j)\}$ is connected, and there is an MBCPP tour (x^1, y^1) that traverses edge (u, v) at least once and visits vertex i. This tour satisfies $x^1_{uv} = 1$. The MBCPP tour (x^2, y^2) obtained from (x^1, y^1) by adding edge (i, j) twice, also satisfies $x^2_{uv} = 1$. After subtracting $ax^1 + by^1 = c$ from $ax^2 + by^2 = c$, we obtain $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E\backslash\{(u, v)\}$.

Let $\mathcal{P}$ be a path joining vertices 1 and u that does not use edge (u, v). The MBCPP tour (x^1, y^1) that traverses the path $\mathcal{P}$ and the edge (u, v) twice satisfies $x^1_{uv} = 1$ and then $ax^1 + by^1 = c$, i.e. $\sum_{e\in\mathcal{P}}(a_e + b_e) + a_{uv} + b_{uv} = c$. Since $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E\backslash\{(u, v)\}$, we obtain $a_{uv} + b_{uv} = c$. The same argument can be applied to deduce that $\sum_{e\in\mathcal{C}} a_e = c$ for any cycle $\mathcal{C}$ containing edge (u, v).

Let $\mathcal{C}$ now be any cycle in graph G that does not contain edge (u, v). There is an MBCPP tour that traverses the edges in $\mathcal{C}$ once and a subset F of other edges in E including (u, v) twice. This tour satisfies $x^1_{uv} = 1$ and then $ax^1 + by^1 = c$, that is, $\sum_{e\in F}(a_e + b_e) + \sum_{e\in\mathcal{C}} a_e = c$. Since $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E\backslash\{(u, v)\}$ and $a_{uv} + b_{uv} = c$, we obtain $\sum_{e\in\mathcal{C}} a_e = 0$ for all cycles $\mathcal{C}$ not containing edge (u, v). By combining the previous results, we obtain $\sum_{e\in\mathcal{C}} b_e = 0$ for any cycle $\mathcal{C}$ in G.

Let $(i, j) \in E$ be an arbitrary edge. There are two edge-disjoint paths $\mathcal{P}_1, \mathcal{P}_2$ joining vertices i and j that do not contain the edge (i, j). As in previous theorems, let us consider the cycles $\mathcal{P}_1 \cup \{(i, j)\}$, $\mathcal{P}_2 \cup \{(i, j)\}$, and $\mathcal{P}_1 \cup \mathcal{P}_2$. Given that $\sum_{e\in\mathcal{C}} b_e = 0$, we obtain $b_{ij} = 0$ for each edge $(i, j) \in E$. Since $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E\backslash\{(u, v)\}$ and $a_{uv} + b_{uv} = c$, we obtain $a_{ij} = 0$ and $a_{uv} = c$ holds. Then, the inequality $ax + by \leq c$ turns out to be $cx_{uv} \leq c$ and $x_{uv} \leq 1$ is facet-inducing for MBCPP(G). □

Theorem 5 *Inequalities (3), $x_{uv} \geq y_{uv}$ for every edge $(u, v) \in E$ are facet-inducing for MBCPP(G) if, and only if, graph $G\backslash\{(u, v)\}$ is 3-edge connected.*

Proof If graph G is 3-edge connected but graph $G\backslash\{(u, v)\}$ is not 3-edge connected, there is at least one cut-set $\delta(S)$ containing exactly the edge (u, v) and two more edges, say f, g. In this case, it can be seen that the inequality $x_{uv} \geq y_{uv}$ is not facet-inducing because it is the sum of two *parity inequalities* (6), which will be presented in Sect. 4.1, associated with $\delta(S)$ and with $F = \{f\}$ and $F = \{g\}$, respectively. The proof that inequalities (3) are facet-inducing when $G\backslash\{(u, v)\}$ is 3-edge connected is similar to those in the previous theorems and is omitted here. □

Theorem 6 *Connectivity inequalities* (2) *are facet-inducing for MBCPP(G) if graph G is 3-edge connected and subgraphs $G(S)$ and $G(V\backslash S)$ are 2-edge connected.*

Proof Let us suppose there is another valid inequality $ax + by \geq c$ such that

$$\left\{(x, y) \in \text{MBCPP}(G) : \sum_{e\in\delta(S)} \left(x_e + y_e\right) - 2x_f = 0\right\} \subseteq \{(x, y) \in \text{MBCPP}(G) : \; ax + by = c\}.$$

Again, we can assume that $c = 0$. As in previous theorems, it can be seen that $a_{ij} + b_{ij} = 0$ for all $(i, j) \in E(V\backslash S) \cup E(S)\backslash\{f\}$. For each edge $(l, m) \in \delta(S)$, we can build an MBCPP tour that uses two copies of each one of the following edges: each edge in a path in $G(V\backslash S)$ from vertex 1 to vertex l, edge (l, m), each edge in a path in $G(S)$ from m to an endpoint of edge f, and edge f. This tour satisfies (2) with equality and, therefore, $a_{lm} + b_{lm} + a_f + b_f = 0$.

Let $\mathcal{C}$ be any cycle in graph $G(V\backslash S)$. It is easy to obtain $\sum_{e\in\mathcal{C}} a_e = \sum_{e\in\mathcal{C}} b_e = 0$. Let $\mathcal{C}$ now be any cycle in graph $G(S)$ that does not contain edge f. Given a vertex i belonging to cycle $\mathcal{C}$ and an endpoint j of edge f, let $\mathcal{P}$ be a path joining vertices $1, i$, and j, such that it traverses $\delta(S)$ exactly once. If we consider two copies of each edge in $\mathcal{C}\cup\mathcal{P}\cup\{f\}$, we obtain an MBCPP tour (x^1, y^1) satisfying $\sum_{e\in\delta(S)} \left(x_e^1+y_e^1\right) = 2x_f^1$. If we remove one copy of each edge in $\mathcal{C}$ from (x^1, y^1), we obtain another MBCPP tour (x^2, y^2) also satisfying $\sum_{e\in\delta(S)} \left(x_e^2 + y_e^2\right) = 2x_f^2$. By subtracting $ax^2 + by^2 = 0$ from $ax^1 + by^1 = 0$, we obtain $\sum_{e\in\mathcal{C}} b_e = 0$ and, hence, $\sum_{e\in\mathcal{C}} a_e = 0$ for all the cycles in graph $G(S)$ that do not contain edge f. Then,

$$\sum_{e\in\mathcal{C}} a_e = 0, \quad \sum_{e\in\mathcal{C}} b_e = 0 \quad \forall \text{ cycle } \mathcal{C} \text{ in graphs } \; G(V\backslash S) \text{ or } G(S)\backslash\{f\}.$$

Let $(i, j) \in E(V\backslash S)$. Given that graph G is 3-edge connected, there are two edge-disjoint paths $\mathcal{P}_1, \mathcal{P}_2$ joining vertices i and j that do not contain the edge (i, j). We consider several cases:

(a) $\mathcal{P}_1$ and $\mathcal{P}_2$ are in $G(V\backslash S)$. In this case, the usual process leads us to $a_{ij} = b_{ij} = 0$.

(b) $\mathcal{P}_1$ is in $G(V\backslash S)$ while $\mathcal{P}_2$ traverses the cut-set $\delta(S)$. We can assume that $\mathcal{P}_2$ traverses $\delta(S)$ exactly twice. Let (x^1, y^1) be the MBCPP tour that uses the edge (i, j) and the path $\mathcal{P}_2$ once, the edges needed to connect $\mathcal{P}_2$ with f if f is not in $\mathcal{P}_2$ twice, and those needed to connect it with the depot twice, if necessary. Then, let (x^2, y^2) be the tour obtained after replacing the edge (i, j) in (x^1, y^1) by the edges in the path $\mathcal{P}_1$. Note that if 1 is in $\mathcal{P}_1$, this could cause some edges to appear three times and (x^2, y^2) would not be an MBCPP tour. In that case, we just remove two copies of these edges. Both tours satisfy inequality (2) with equality and, by comparing them, we obtain $a_{ij} = b_{ij} = 0$.

(c) Both $\mathcal{P}_1$ and $\mathcal{P}_2$ traverse the cut-set $\delta(S)$ (exactly twice). We are going to see that we can use them to construct two edge-disjoint paths joining i and j, where one of them is in $G(V\backslash S)$, and so we will be in the same situation as in (b). Since graph $G(V\backslash S)$ is 2-edge connected, there is a path $\mathcal{P}_3$ in $G(V\backslash S)$ also joining i

and j and not containing (i, j). If $\mathcal{P}_3$ and $\mathcal{P}_1$ (or $\mathcal{P}_2$) are edge-disjoint paths, we are done. Otherwise, it can be seen that it is possible to modify $\mathcal{P}_3$ to construct a new path $\mathcal{P}_4$ consisting of three parts, P_i, P_m and P_j, where P_i (starting at vertex i) satisfies $P_i \subset \mathcal{P}_1$ or $P_i \subset \mathcal{P}_2$, P_j (ending at vertex j) satisfies $P_j \subset \mathcal{P}_1$ or $P_j \subset \mathcal{P}_2$, and P_m contains only edges from $\mathcal{P}_3$ and is edge-disjoint with $\mathcal{P}_1 \cup \mathcal{P}_2$. Let us assume that $P_i \subset \mathcal{P}_1$. If $P_j \subset \mathcal{P}_1$, then $\mathcal{P}_4$ and $\mathcal{P}_2$ are edge-disjoint paths and we are done. Otherwise, using the subpath of $\mathcal{P}_2$ leaving vertex i and the subpath of $\mathcal{P}_1$ reaching vertex j, we can construct a new path $\mathcal{P}_5$, traversing exactly twice the cut-set $\delta(S)$, which is edge-disjoint with $\mathcal{P}_4$.

Similarly, for each edge $(i, j) \in E(S) \backslash \{f\}$, we obtain $a_{ij} = b_{ij} = 0$.

Let us denote the edges in $\delta(S)$ as $e_1, \ldots, e_p$, where $p \geq 3$ since graph G is 3-edge connected. Now consider two edges $e_1, e_2 \in \delta(S)$. Given that graphs $G(S)$ and $G(V \backslash S)$ are connected, there is an MBCPP tour (x^1, y^1) using the edges in a path that starts at the depot, traverses e_1 and ends at edge f twice. Let (x^2, y^2) be the MBCPP tour obtained from (x^1, y^1) after replacing the second traversal of e_1 with the first traversals of the edges in a path joining the endpoints of e_1, using e_2, and not traversing f. Again, if some edges appear three times, two copies of them are removed. After subtracting $ax^1 + by^1 = 0$ from $ax^2 + by^2 = 0$, we obtain $b_{e_1} = a_{e_2}$. If we interchange the roles of the edges e_1 and e_2, we obtain $b_{e_2} = a_{e_1}$. Proceeding in this way with all the pairs of edges in $\delta(S)$, we obtain $a_{e_i} = b_{e_j}$ for all $i \neq j \in \{1, \ldots, p\}$ and then $a_{e_i} = a_{e_j} = b_{e_i} = b_{e_j}$ for all i, j (because $p \geq 3$ holds).

Finally, given that graph $G(S)$ is 2-edge connected, there is a cycle $\mathcal{C}$ in graph $G(S)$ that contains the edge f. Let (x^1, y^1) be an MBCPP tour using the edges in $\mathcal{C}$ once and the edges in a path $\mathcal{P}$ crossing $(S : V \backslash S)$ and joining the depot 1 to a vertex i belonging to the cycle $\mathcal{C}$ twice. Let (x^2, y^2) be the MBCPP tour using all the edges in $\mathcal{P}$ twice, the edges in the path formed with the edges in the cycle $\mathcal{C}$ from vertex i to an endpoint of edge f twice plus the edge f twice. Both MBCPP tours satisfy $\sum_{e \in \delta(S)} \left(x_e + y_e\right) = 2x_f$ and, after subtracting $ax^2 + by^2 = 0$ from $ax^1 + by^1 = 0$, we obtain $b_f = 0$. Given that $a_{lm} + b_{lm} + a_f + b_f = 0$ holds for any edge $(l, m) \in \delta(S)$, we obtain $a_f = -2a_{lm}$ and hence the connectivity inequality (2) is facet-inducing for MBCPP(G). □

Note The above result is also true when $V \backslash S$ contains only one edge or only the depot, and when S contains only edge f, as long as G is 3-edge connected.

4 Other inequalities

In this section we present several new families of valid inequalities for the MBCPP, namely parity, K-C and p-connectivity inequalities, and we study the conditions under which they are facet-inducing for MBCPP(G).

4.1 Parity inequalities

Constraints (1) are not linear inequalities. In order to force the solution to satisfy these parity constraints, we can use other linear inequalities as the set-parity inequalities proposed in [2] for the Prize-Collecting Rural Postman Problem (PRPP) which, as previously mentioned, is a special case of the MBCPP. A first version of these inequalities was proposed in [3], and later corrected in [2]. They are based on the so-called co-circuit inequalities proposed by Barahona and Grötschel [5] for the binary matroid problem, and are as follows. Given a vertex set $S \subset V\backslash\{1\}$ and two edge sets $F \subseteq \delta(S)$ and $L \subseteq F$, such that $|F| + |L|$ is odd, then the set-parity inequality is

$$x(\delta(S)\backslash F) + y(F\backslash L) \geq x(F) + y(L) - (|F| + |L|) + 1. \tag{5}$$

It is easy to see that inequalities (5) are valid for the MBCPP, but we failed to prove that they induce facets of MBCPP(G). However, we found the following ones, which we will call parity inequalities, that dominate inequalities (5) and are facet-inducing for MBCPP(G):

$$(x - y)(\delta(S)\backslash F) \geq (x - y)(F) - |F| + 1, \quad \forall S \subset V, \quad \forall F \subset \delta(S) \text{ with } |F| \text{ odd}. \tag{6}$$

Theorem 7 *Parity inequalities (6) are valid for MBCPP(G).*

Proof Let (x^*, y^*) be an MBCPP tour. We have to prove that $(x^* - y^*)(\delta(S)\backslash F) \geq (x^* - y^*)(F) - |F| + 1$. If $(x^* - y^*)(F) \leq |F| - 1$, this inequality reduces to $(x^* - y^*)(\delta(S)\backslash F) \geq 0$, which is obviously satisfied. Let us suppose then that $(x^* - y^*)(F) = |F|$, which is only satisfied when $x^*(F) = |F|$ and $y^*(F) = 0$. In this case, the inequality becomes $(x^* - y^*)(\delta(S)\backslash F) \geq 1$. Since cut-set $(S : V\backslash S)$ must be traversed an even number of times and $|F|$ is odd, $\delta(S)\backslash F$ must be traversed an odd number of times. Given that $x^*_e \geq y^*_e$ for any edge e, $x^*_e - y^*_e = 1$ must hold for at least one edge $e \in (\delta(S)\backslash F)$. □

Theorem 8 *Parity inequalities (6) are stronger than set-parity inequalities* (5).

Proof Let $S \subset V\backslash\{1\}$, $F \subseteq \delta(S)$, $L \subseteq F$, $|F| + |L|$ odd. Define $F' = F\backslash L$. Since $|F| + |L|$ is an odd number, $|F'|$ is also odd. We will prove that the parity inequality (6) associated with S and F' dominates the cocircuit inequality (5) associated with sets S, F and L.

Given that $F' = F\backslash L$, then $\delta(S)\backslash F' = (\delta(S)\backslash F) \cup L$ and the inequality (6) associated with sets S and F' can be written as:

$$x(\delta(S)\backslash F) + x(L) - y(\delta(S)\backslash F) - y(L) \geq x(F) - x(L) - y(F) \\ + y(L) - (|F| - |L|) + 1,$$

or equivalently

$$x(\delta(S)\backslash F) + y(F) - y(L) \geq x(F) + y(L) - 2x(L) \\ + y(\delta(S)\backslash F) - (|F| - |L|) + 1,$$

which can be written as

$$x(\delta(S)\backslash F) + y(F\backslash L) \geq x(F) + y(L) - (|F| + |L|) + 1 + 2|L| - 2x(L)$$
$$+y(\delta(S)\backslash F) = x(F) + y(L) - (|F| + |L|) + 1 + 2(|L| - x(L))$$
$$+y(\delta(S)\backslash F),$$

whose RHS is obviously greater than or equal to the RHS in the set-parity inequality. □

Note 1 Before proving that parity inequalities induce facets of MBCPP(G), in what follows we will describe two types of MBCPP tours satisfying (6) with equality. Given a graph $G = (V, E)$ and $T \subset V$, with $|T|$ even, recall that a subset of edges $E' \subset E$ is a T-join if, in the subgraph $G' = (V, E')$, the degree of v is odd if and only if $v \in T$ (see [17], for instance). If G is connected, it has a T-join for each set $T \subset V$ with $|T|$ even.

Type 1 Let us consider the cut-set depicted in Fig. 1a, with $|F| = 3$, where we assume that $G(S)$ and $G(V\backslash S)$ are connected. Each MBCPP solution traversing all the edges in F has to traverse the cut-set $(V\backslash S, S)$ at least one more time. Let e' be either an edge in $\delta(S)\backslash F$ or a copy of an edge in F. Figure 1b shows this second case. Let $T \subset V\backslash S$ be the set of vertices incident with an odd number of edges in $F \cup \{e'\}$. Given that $|F \cup \{e'\}|$ is even, $|T|$ is also even and there is a T-join E' in $G(V\backslash S)$. This same process is done in $G(S)$ (see Fig. 1c). The subgraph in $G(V\backslash S)$, G^*, induced by the edges in E', the vertices incident with $F \cup \{e'\}$, and the depot can be disconnected. G^* can be transformed into a connected graph (see Fig. 1d) by adding two copies of some edges connecting its components (and any other vertex i as needed in the proof of Theorem 9). Then, the two subgraphs built in $G(V\backslash S)$ and $G(S)$, plus the edges in $F \cup \{e'\}$, define an MBCPP tour that satisfies (6) with equality.

Note that the above procedure can also be used if, in addition to the edges in $F \cup \{e'\}$, we also consider two copies of any q edges in $\delta(S)\backslash F$. In this case we would obtain:

$$x(F) = |F|,\ y(F) = 0,\ x(\delta(S)\backslash F) = 1 + q \text{ and } y(\delta(S)\backslash F) = 0 + q,$$
$$\text{if } e' \in \delta(S)\backslash F, \text{ or}$$
$$x(F) = |F|,\ y(F) = 1,\ x(\delta(S)\backslash F) = 0 + q \text{ and } y(\delta(S)\backslash F) = 0 + q, \quad \text{if } e' \in F.$$

In both cases, the MBCPP tours satisfy (6) with equality.

Type 2 Consider now all the edges in F, except one of them, and two copies of any q edges in $\delta(S)\backslash F$. From these $|F| - 1 + 2q$ edges we can apply the above procedure to obtain an MBCPP tour satisfying:

$$x(F) = |F| - 1,\ y(F) = 0,\ x(\delta(S)\backslash F) = 0 + q \text{ and } y(\delta(S)\backslash F) = 0 + q,$$

and, therefore, satisfying (6) with equality.

Theorem 9 *Parity inequalities (6) are facet-inducing for MBCPP(G) if graph G is 3-edge connected and graphs $G(S)$ and $G(V\backslash S)$ are 2-edge connected.*

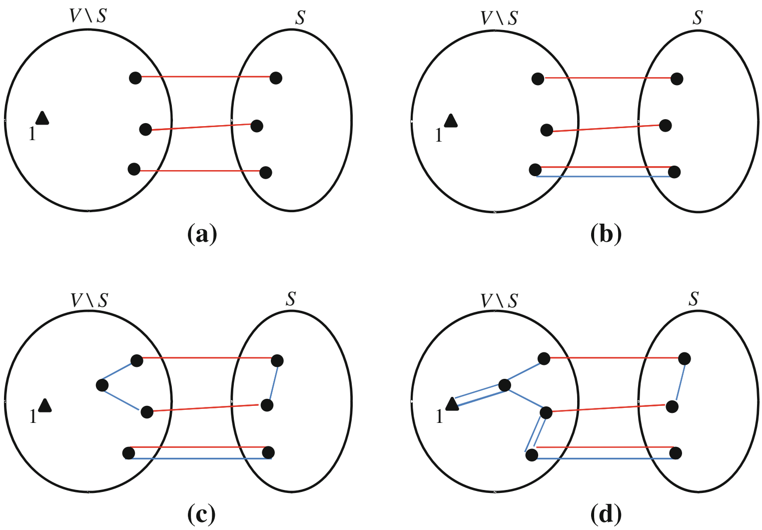

Fig. 1 Construction of MBCPP tours of type 1 satisfying (6) with equality

Proof Inequalities (6) can be written in the following form:

$$(x - y)(F) - (x - y)(\delta(S)\setminus F)) \leq |F| - 1. \tag{7}$$

Let us suppose there is another valid inequality $ax + by \leq c$ such that

$$\{(x, y) \in \text{MBCPP}(G) : (x - y)(F) - (x - y)(\delta(S)\setminus F)) = |F| - 1\}$$
$$\subseteq \{(x, y) \in \text{MBCPP}(G) : \quad ax + by = c\}.$$

Let $(i, j) \in E(S) \cup E(V\setminus S)$. Given that G is 3-edge connected, graph $G\setminus\{(i, j)\}$ is connected, and there is an MBCPP tour (x^1, y^1) that satisfies (7) with equality and visits vertex i (see Note 1). The MBCPP tour (x^2, y^2) obtained from (x^1, y^1) by adding the traversal of edge (i, j) twice, also satisfies (7) with equality. Then $ax^1 + by^1 = c$, $ax^2 + by^2 = c$ and, after subtracting the first expresion from the second one, we obtain

$$a_{ij} + b_{ij} = 0 \quad \forall (i, j) \in E(S) \cup E(V\setminus S).$$

Let $(i, j) \in E(V\setminus S)$. Given that G is 3-edge connected and $G(V\setminus S)$ is 2-edge connected, we can construct, as in the proof of theorem 6, two edge-disjoint paths $\mathcal{P}_1$ and $\mathcal{P}_2$ joining vertices i and j that do not contain the edge (i, j), such that at least one of them is in $G(V\setminus S)$.

If both paths are in $G(V\setminus S)$, proceeding as in Note 1, we can build an MBCPP tour (x^1, y^1) in G satisfying (7) with equality such that it uses edge (i, j) exactly once. To do that, the "parity" label of vertices i and j is switched before the T-join is computed and then the edge (i, j) is added. We define three more MBCPP tours in the following way. Consider (x^1, y^1) and suppose we add one copy of each edge in paths $\mathcal{P}_1, \mathcal{P}_2$. The resulting tour is even and connected, but it is not necessarily an MBCPP tour because some edges can appear three times. If we remove two copies of each one of these edges used three times, we obtain an even and connected tour (x^2, y^2) that is a feasible solution for the MBCPP. Let (x^3, y^3) be the tour obtained from (x^1, y^1) after removing the edge (i, j), adding one copy of each edge in path $\mathcal{P}_1$, and then removing two copies of each one of these edges used three times. Finally, let (x^4, y^4) be the tour obtained from (x^1, y^1) after removing the edge (i, j), adding one copy of each edge that is in path $\mathcal{P}_2$, and then removing two copies of each one of these edges used three times. All these four MBCPP tours satisfy (7) with equality and then also satisfy $ax + by = c$.

Let us define $\alpha(\mathcal{P}_i) = \sum_{e \in P_i^1} a_e + \sum_{e \in P_i^2} b_e$, where P_i^1 is the set of edges in path $\mathcal{P}_i$ that are traversed only once in (x^1, y^1) and P_i^2 the set of edges in path $\mathcal{P}_i$ that are not traversed at all or are traversed twice in (x^1, y^1). If we subtract the expression $ax^1 + by^1 = c$ from $ax^2 + by^2 = c$, we obtain $\alpha(\mathcal{P}_1) + \alpha(\mathcal{P}_2) = 0$. In the same way, by comparing tours 3 and 4 above, we obtain $\alpha(\mathcal{P}_1) = \alpha(\mathcal{P}_2)$ and then $\alpha(\mathcal{P}_1) = \alpha(\mathcal{P}_2) = 0$. Finally, if we compare tours 1 and 3, we obtain $a_{ij} = \alpha(\mathcal{P}_1)$ and then $a_{ij} = 0$. Since $a_{ij} + b_{ij} = 0$, also $b_{ij} = 0$. For each edge $(i, j) \in E(S)$, a similar process leads to $a_{ij} = b_{ij} = 0$.

Let us now suppose that path $\mathcal{P}_2$ is not in $G(V \setminus S)$, i.e. it leaves the graph $G(V \setminus S)$ and traverses the cut-set $\delta(S)$. Given that graph $G(S)$ is connected, we can assume that path $\mathcal{P}_2$ traverses the cut-set $\delta(S)$ exactly once in each direction through two edges, say e_1 and e_2. We consider three cases:

(1) $e_1, e_2 \in F$. Let (x^1, y^1) be the MBCPP tour of type 2, satisfying (7) with equality, which traverses (i, j) and all the edges in $F \setminus \{e_2\}$ once and does not traverse e_2. It can be seen that three MBCPP tours (x^2, y^2), (x^3, y^3) and (x^4, y^4) as defined above would also satisfy (7) with equality. Note that when we add one copy of each edge to path $\mathcal{P}_2$, we obtain an MBCPP tour of type 1 that uses each edge in $F \setminus \{e_1\}$ exactly once and edge e_1 twice.

(2) $e_1, e_2 \notin F$. Let (x^1, y^1) be the MBCPP tour of type 1, satisfying (7) with equality, which traverses (i, j) and all the edges in $F \cup \{e_1\}$ once and does not traverse e_2. Again, three MBCPP tours (x^2, y^2), (x^3, y^3) and (x^4, y^4) as defined above would also satisfy (7) with equality. Note that when we add one copy of each edge to path $\mathcal{P}_2$, we obtain an MBCPP tour of type 1 that uses each edge in $F \cup \{e_2\}$ exactly once and the edge e_1 twice.

(3) $e_1 \in F, e_2 \notin F$. Let (x^1, y^1) be the MBCPP tour of type 2 satisfying (7) with equality, traversing (i, j) and all the edges in $F \setminus \{e_1\}$ once and not traversing e_2. Again, the three MBCPP tours (x^2, y^2), (x^3, y^3) and (x^4, y^4) as defined above would also satisfy (7) with equality. Note that when we add one copy of each edge to path $\mathcal{P}_2$, we obtain an MBCPP tour of type 1 that uses each edge in $F \cup \{e_2\}$ exactly once.

In any of the 3 cases above, following a similar reasoning to that of the case in which path $\mathcal{P}_2$ is in $G(V \setminus S)$, we obtain $a_{ij} = b_{ij} = 0$ for all edges $(i, j) \in E(V \setminus S)$. The same result can be proved for any edge $(i, j) \in E(S)$.

Consider now an edge $(i, j) \in \delta(S)$. As we have seen in Note 1, there is an MBCPP tour (x^1, y^1) that satisfies (7) with equality and does not use (i, j), while visiting vertices i and j. Let (x^2, y^2) be the tour obtained after adding edge (i, j)

twice to (x^1, y^1). Since $ax^1 + by^1 = ax^2 + by^2 = c$, subtracting these expressions, we obtain $a_{ij} + b_{ij} = 0$ for all $(i, j) \in \delta(S)$.

Let $e_1, e_2 \in F$. Let (x^1, y^1) be the MBCPP tour that uses all the edges in $F \backslash \{e_1\}$ exactly once and edge e_1 twice and let (x^2, y^2) be the tour that uses all the edges in $F \backslash \{e_2\}$ exactly once and edge e_2 twice. Both tours can be constructed satisfying (7) with equality. By comparing them, and considering that $a_{ij} = b_{ij} = 0$ for all edges $(i, j) \in E(S) \cup E(V \backslash S)$, we obtain $b_{e_1} = b_{e_2}$ and, therefore, $a_{e_1} = a_{e_2}$. By iterating this argument, we obtain $a_{ij} = \lambda$ and $b_{ij} = -\lambda$ for all $(i, j) \in F$.

For each $e_1 \in \delta(S) \backslash F$, consider any $e_2 \in F$. Let (x^1, y^1) be the MBCPP tour that uses all the edges in $F \cup e_1$ exactly once and let (x^2, y^2) be the tour that uses all the edges in $F \backslash \{e_2\}$ exactly once and edge e_2 twice. By comparing them, we obtain $a_{e_1} = b_{e_2} = -\lambda$ and, therefore, $b_{e_1} = \lambda$.

Then, inequality $ax + by \le c$ reduces to $\lambda x(F) - \lambda y(F) - \lambda x(\delta(S) \backslash F) + \lambda y(\delta(S) \backslash F) \le c$. Given that the MBCPP tour (x^1, y^1) above, for example, satisfies this inequality with equality, we obtain $\lambda |F| - \lambda = c$ and, hence, inequality $ax + by \le c$ reduces to $x(F) - y(F) - x(\delta(S) \backslash F) + y(\delta(S) \backslash F) \le |F| - 1$. □

4.2 K-C inequalities

K-C inequalities [10] are a well-known family of facet-inducing inequalities for the Rural Postman Problem (RPP) and many other arc routing problems. In this section we show that these inequalities can be transformed in order to obtain new valid and facet-inducing inequalities for MBCPP(G) that we will continue to call K-C inequalities for the sake of simplicity.

Let $\{M_0, \ldots, M_K\}$, with $K \ge 3$, be a partition of V, where for instance $1 \in M_0 \cup M_K$. Given an edge $e_i \in E(M_i)$ for each $i = 1, \ldots, K-1$, and a subset of edges $F \subseteq (M_0 : M_K)$ with $|F|$ even, the K-C inequalities for the MBCPP are defined as:

$$\begin{aligned} &(K-2)(x-y)\Big((M_0 : M_K) \backslash F\Big) - (K-2)(x-y)(F) \\ &\quad + \sum_{\substack{0 \le i < j \le K \\ (i,j) \ne (0,K)}} \Big((j-i)x(M_i : M_j) + (2-j+i)y(M_i : M_j)\Big) \\ &\quad -2\sum_{i=1}^{K-1} x_{e_i} \ge -(K-2)|F| \end{aligned} \tag{8}$$

The coefficients and structure of the K-C inequalities are shown in Fig. 2. Edges in F are represented by thick lines. For each pair (a, b) associated with an edge e, a and b represent the coefficients of x_e and y_e, respectively. Note that if a solution traverses each edge in F and each edge e_i exactly once, it has to satisfy

$$\begin{aligned} &(K-2)(x-y)\Big((M_0 : M_K) \backslash F\Big) \\ &\quad + \sum_{\substack{0 \le i < j \le K \\ (i,j) \ne (0,K)}} \Big((j-i)x(M_i : M_j) + (2-j+i)y(M_i : M_j)\Big) \ge 2(K-1), \end{aligned}$$

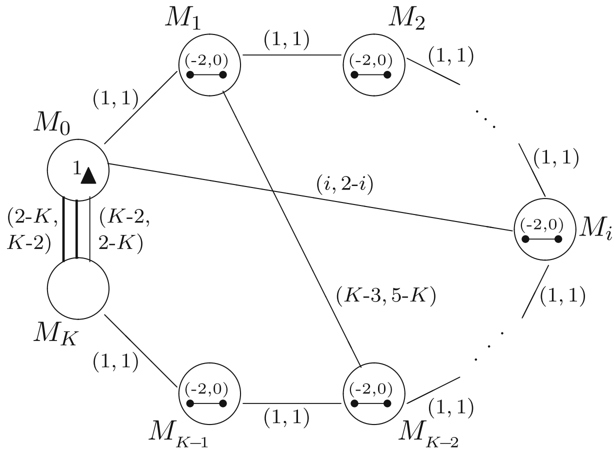

Fig. 2 Coefficients of the K-C inequality

which, as far as the x variables are concerned, resembles the version of the K-C inequality for the undirected RPP [10], as well as the "switched" K-C inequalities presented in [21].

If the depot $1 \notin M_0 \cup M_K$ but $1 \in M_d$, for a $d \in \{1, \ldots, K-1\}$, and $|F| \geq 2$, the corresponding K-C inequality is

$$(K-2)(x-y)\Big((M_0 : M_K)\backslash F\Big) - (K-2)(x-y)(F) + \sum_{\substack{0\leq i<j\leq K \\ (i,j)\neq(0,K)}} \Big((j-i)x(M_i : M_j) + (2-j+i)y(M_i : M_j)\Big) -2\sum_{\substack{i=1 \\ i\neq d}}^{K-1} x_{e_i} \geq -(K-2)|F| + 2. \tag{9}$$

Note that when $|F| = 0$ or $K = 2$, the K-C inequality (9) is not valid since (0, 0) is a feasible solution while the RHS is 2. Moreover, when $K = 2$, the K-C inequality (8) reduces to a connectivity inequality (2).

Theorem 10 *K-C inequalities* (8) *and* (9) *are valid for MBCPP(G).*

Proof We will prove only the validity of inequalities (8) since the proof for inequalities (9) is analogous. Let (x^*, y^*) be an MBCPP tour. If $x^*_{e_j} = 0$ for a $j \in \{1, \ldots, K-1\}$, then we could consider a new K-C configuration with $K-1$ subsets where M_j and M_{j+1} have been merged into a single subset $M_j \cup M_{j+1}$, and it can be proved that if its associated K-C inequality is satisfied by (x^*, y^*), then the original K-C inequality is also satisfied by (x^*, y^*). Then, in what follows, we can assume that $x^*_{e_1} = \cdots = x^*_{e_{K-1}} = 1$ and we have to prove that

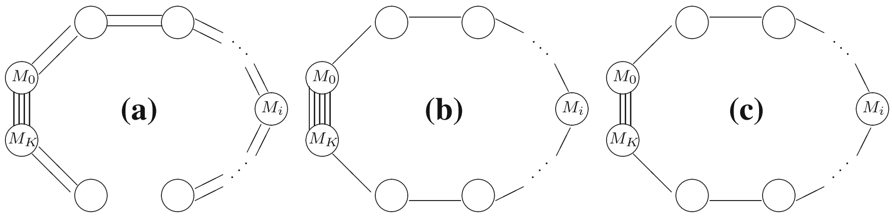

Fig. 3 MBCPP solutions used in the proof of Theorem 10

$$(K-2)(x^*-y^*)\Big((M_0:M_K)\backslash F\Big)-(K-2)(x^*-y^*)(F) + \sum_{\substack{0\le i<j\le K\\(i,j)\neq(0,K)}}\Big((j-i)x^*(M_i:M_j)+(2-j+i)y^*(M_i:M_j)\Big) \geq 2(K-1)-(K-2)|F|. \quad (10)$$

Let us suppose first that $(x^*-y^*)(F)=|F|$. This means that (x^*, y^*) traverses each edge in F once. In this case, inequality (10) reduces to

$$(K-2)(x^*-y^*)\Big((M_0:M_K)\backslash F\Big) + \sum_{\substack{0\le i<j\le K\\(i,j)\neq(0,K)}}\Big((j-i)x^*(M_i:M_j)+(2-j+i)y^*(M_i:M_j)\Big)\geq 2(K-1).$$

It is easy to see that this inequality holds since (x^*, y^*) starts at M_0, visits all the subsets $M_1,\ldots,M_{K-1}$ and returns to M_0 (see Fig. 3a, b), where edges in F used by the tour are depicted in bold).

Consider now that $(x^*-y^*)(F)=|F|-1$, that is, (x^*, y^*) traverses the edges in F an odd number of times. In this case, inequality (10) reduces to

$$(K-2)\big(x^*-y^*\big)\Big((M_0:M_K)\backslash F\Big) + \sum_{\substack{0\le i<j\le K\\(i,j)\neq(0,K)}}\Big((j-i)x^*(M_i:M_j)+(2-j+i)y^*(M_i:M_j)\Big)\geq K.$$

The best way of starting at M_0, visiting $M_1,\ldots,M_{K-1}$ and ending at M_K (remember that the edges in F are traversed an odd number of times) gives a left-hand side value greater than or equal to K (see Fig. 3c), and the previous inequality holds.

All the other cases when $(x^*-y^*)(F)<|F|-1$ can be proved easily since, in this case, the right-hand side of the inequality is 2 or less, which should be clearly satisfied by any feasible solution. □

Theorem 11 *K-C inequalities* (8) *and* (9) *are facet-inducing for MBCPP(G) if graph G is 3-edge connected, graph $G(M_i)$ is 3-edge connected for $i=0,\ldots,K$, $|(M_i:M_{i+1})|\geq 2$ for $i=0,\ldots,K-1$, and $|F|\geq 2$.*

Proof K-C inequalities have been proved to be facet-defining for other arc routing problems. Therefore, we will omit the proof for this new variant for the sake of brevity. □

4.3 p-connectivity inequalities

Consider an MBCPP instance defined on the complete graph with 5 vertices, K_5. Consider the fractional solution shown in Fig. 4a, where a double solid (dotted) line means that both variables x and y take value 1 (0.5). It can be seen that it satisfies all the inequalities presented in previous sections. In particular, it satisfies the connectivity inequality (2) associated with set $S_1 = \{2, 3\}$ and $f_1 = (2, 3)$, as well as that associated with $S_2 = \{4, 5\}$ and edge $f_2 = (4, 5)$.

Note, however, that sets S_1 and S_2 are not properly connected to the depot. As we will show later, the following inequality is valid for the MBCPP and cuts fractional solutions like the one depicted in Fig. 4a:

$$(x + y)(\delta(\{1\}) + 2x(S_1 : S_2) \geq 2x_{23} + 2x_{45}.$$

The above inequality can be extended as follows. Let $\{S_0, S_1, S_2\}$ be a partition of V and assume that $1 \in S_0$. Let $e_1 \in E(S_1)$ and $e_2 \in E(S_2)$. The following inequality will be referred to as a 2-connectivity inequality:

$$(x + y)(\delta(S_0)) + 2x(S_1 : S_2) \geq 2x_{e_1} + 2x_{e_2}.$$

This inequality is represented in Fig. 4b, where for each pair (a, b) associated with an edge e, a and b represent the coefficients of x_e and y_e, respectively. A different version of these 2-connectivity inequalities is obtained when the depot is not in S_0. If, for example, $1 \in S_1$ (see Fig. 4c), the inequality takes the form:

$$(x + y)(\delta(S_0)) + 2x(S_1 : S_2) \geq 2x_{e_0} + 2x_{e_2},$$

where e_0 is a given edge in S_0.

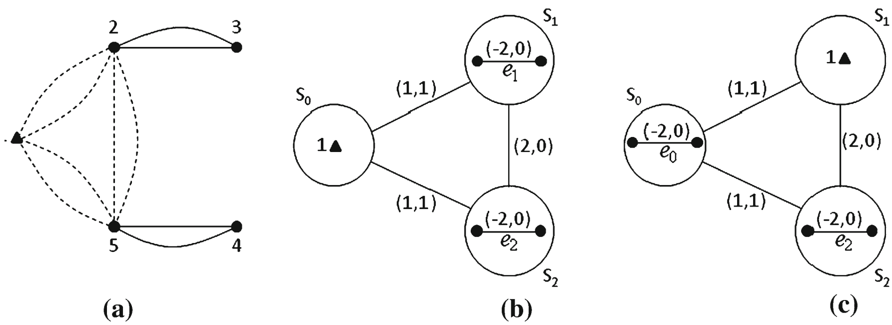

Fig. 4 2-connectivity inequalities

2-connectivity inequalities can be generalized by considering any number $p+1$ of sets. Let $\{S_0, \dots, S_p\}$ be a partition of V. Assume that $1 \in S_d$, $d \in \{0, \dots, p\}$ and consider one edge $e_j \in E(S_j)$ for every $j \in \{0, \dots, p\} \backslash \{d\}$. The following inequality

$$(x+y)(\delta(S_0)) + 2 \sum_{1 \leq r < t \leq p} x(S_r : S_t) \geq 2 \sum_{i=0, i \neq d}^{p} x_{e_i} \tag{11}$$

is valid and will be referred to as a p-connectivity inequality.

Theorem 12 *p-connectivity inequalities* (11) *are valid for MBCPP(G).*

Proof For the sake of simplicity we will assume that $1 \in S_0$. Let (x^*, y^*) be an MBCPP tour. We have to prove that $(x^*+y^*)(\delta(S_0)) + 2\sum_{1 \leq r<t \leq p} x^*(S_r : S_t) \geq 2\sum_{i=1}^{p} x^*_{e_i}$.

Note that if $x^*_{e_j} = 0$ for certain $j \in \{1, \dots, p\}$, then we could consider a new p-connectivity configuration with $p-1$ vertices where S_j and S_{j+1} have been merged into a single vertex $S_j \cup S_{j+1}$. It can be proved that if its associated $(p-1)$-connectivity inequality is satisfied by (x^*, y^*), then the original p-connectivity inequality is also satisfied by (x^*, y^*). Then, in what follows, we can assume that $x^*_{e_1} = \dots = x^*_{e_p} = 1$.

Similarly, if $x^*_e = 1$ for certain $e \in (S_r, S_t)$ with $1 \leq r < t \leq p$, we can define a new partition with $p-1$ elements where S_r and S_t have been merged into $S'_r = S_r \cup S_t$ and where $e'_r = e_r$. Again, if its associated $(p-1)$-connectivity inequality is satisfied by (x^*, y^*), then the original p-connectivity inequality is also satisfied by (x^*, y^*). Hence, we can assume that $(x^*+y^*)(S_i, S_j) = 0$ for any $i, j \in \{1, \dots, p\}$.

Therefore, since $x^*_{e_i} = 1$, $(x^*+y^*)(S_0, S_i) \geq 2$ for each i, and the inequality holds. □

Theorem 13 *p-connectivity inequalities* (11) *are facet-inducing for MBCPP(G) if graph G is 3-edge connected, subgraphs $G(S_i)$, $i = 1, \dots, p$, are 3-edge connected, $|(S_0 : S_i)| \geq 2$, $\forall\, i = 1, \dots, p$, and the graph induced by $V \backslash S_0$ is connected.*

Proof We will assume that $1 \in S_0$. The case $1 \in S_i$, $i \neq 0$, is similar and the proof is omitted here. Let us suppose there is another valid inequality $ax + by \geq c$ such that

$$\left\{(x, y) \in \text{MBCPP}(G) : (x+y)(\delta(S_0)) + 2 \sum_{1 \leq r<t \leq p} x(S_r : S_t) - 2\sum_{i=1}^{p} x_{e_i} = 0\right\}$$
$$\subseteq \{(x, y) \in \text{MBCPP}(G) : ax + by = c\},$$

where, w.l.o.g., we can assume that $c = 0$.

Since subgraphs $G(S_i)$ are 3-edge connected, similar arguments to those used in the proof of Theorem 9 apply to proving that $a_{uv} = b_{uv} = 0$ for every edge $(u, v) \in E(S_i) \backslash \{e_i\}$ for all $i = 0, \dots, p$, and that $b_{e_i} = 0$ for all $i = 1, \dots, p$.

Let S_i and S_j, $i, j \neq 0$ be two sets such that there is an edge $e_{ij} \in (S_i : S_j)$. Since $(S_0 : S_i) \neq \emptyset$, we can construct the tour that traverses both an edge $f \in (S_0 : S_i)$ and an edge e_i twice, which satisfies the inequality (11) as an equality. If we compare this tour with the empty tour, we obtain $a_f + b_f + a_{e_i} = 0$. This result also holds for set

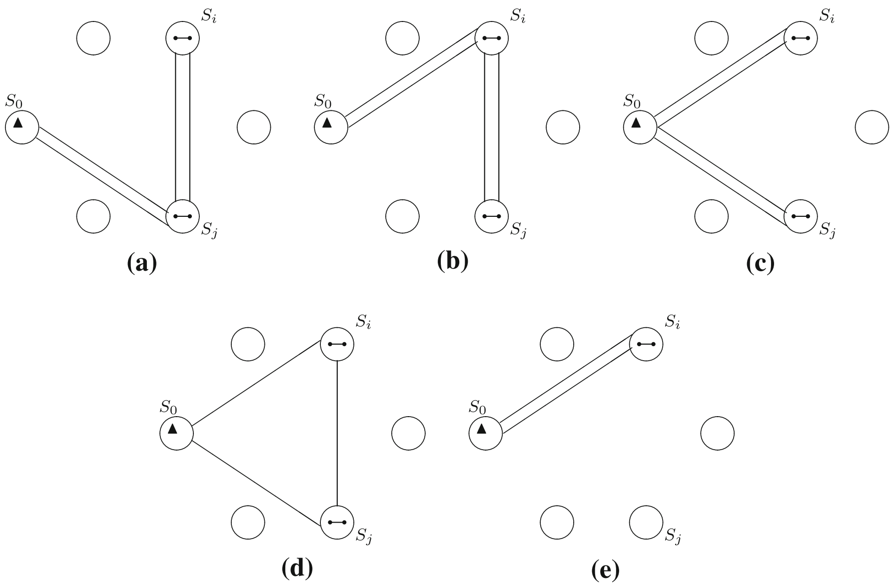

Fig. 5 MBCPP solutions satisfying (11) with equality

S_j. We now construct three tours such as those depicted in Fig. 5a–c) satisfying (11) with equality. Comparing them, we conclude $a_{e_{0i}} + b_{e_{0i}} = a_{e_{ij}} + b_{e_{ij}} = a_{e_{0j}} + b_{e_{0j}} = -a_{e_i} = -a_{e_j}$, for all edges $e_{0i} \in (S_0 : S_i)$ and $e_{0j} \in (S_0 : S_j)$. Given that the graph induced by $V \backslash S_0$ is connected, we can iterate this argument to conclude that

$$a_{e_{uv}} + b_{e_{uv}} = -a_{e_i} = 2\lambda$$

for every edge $e_{uv} \in (S_u : S_v)$ and for every edge e_i, $i = 1, \ldots, p$.

Let $e_{0i}, e'_{0i} \in (S_0 : S_i)$. Since $G(V \backslash S_0)$ is connected, there is a subset S_j such that $(S_i : S_j) \neq \emptyset$. Let $e_{ij} \in (S_i : S_j)$ and $e_{0j} \in (S_0 : S_j)$. Then, we can construct an MBCPP tour traversing edges e_{0i}, e_{ij}, e_{0j}, e_i, and e_j once, and a second tour traversing edges e'_{0i}, e_{ij}, e_{0j}, e_i, and e_j once (see Fig. 5d). Comparing both tours, we obtain $a_{e_{0i}} = a_{e'_{0i}}$ and, since $a_{e_{0i}} + b_{e_{0i}} = a_{e'_{0i}} + b_{e'_{0i}}$, $b_{e_{0i}} = b_{e'_{0i}}$. Now we consider two new MBCPP tours (see Fig. 5e), the first one using e_{0i} twice and e_i once, and the second one using e_{0i}, e'_{0i}, and e_i once. Comparing them, we obtain $b_{e_{0i}} = a_{e'_{0i}}$. Thus, $a_{e_{0i}} = a_{e'_{0i}} = b_{e_{0i}} = b_{e'_{0i}} = \lambda$ for any edges $e_{0i}, e'_{0i} \in (S_0 : S_i)$, $i = 1, \ldots, p$.

As above, let S_i and S_j, $i, j \neq 0$ be two sets such that there is an edge $e_{ij} \in (S_i : S_j)$. The tour (see Fig. 5d) that traverses edge e_{ij} once, one edge $e_{0i} \in (S_0 : S_i)$, one edge $e_{0j} \in (S_0 : S_j)$, and edges e_i and e_j satisfies inequality (11) with equality. Therefore, it satisfies $a_{e_{ij}} + a_{e_{0i}} + a_{e_{0j}} + a_{e_i} + a_{e_j} = a_{e_{ij}} + \lambda + \lambda - 2\lambda - 2\lambda = 0$, and consequently $a_{e_{ij}} = 2\lambda$, which implies $b_{e_{ij}} = 0$.

Finally, since the right-hand side c is 0, dividing the inequality by λ we get the coefficients of the p-connectivity inequality and the proof is completed. □

5 Branch-and-cut algorithm for the MBCPP

The branch-and-cut method presented here is based on a cutting-plane procedure that identifies violated inequalities of the classes described in the previous sections.

The initial LP relaxation contains inequalities (3), which guarantee that a second traversal of an edge can only occur when it has been traversed previously, the bounds on the variables, and a parity inequality (6) with $F = \delta(v)$ for each odd degree vertex v.

5.1 Separation algorithms

In this section, we present the separation algorithms that have been used to identify the inequalities that are violated by the current LP solution at any iteration of the cutting-plane algorithm. Given a fractional solution (x^*, y^*), we will use two support graphs, G^+ and G^-, which are the graphs induced by the edges $e \in E$ with $x_e^* + y_e^* > 0$ and $x_e^* - y_e^* > 0$, respectively, plus the depot vertex, if necessary.

5.1.1 Separation of connectivity inequalities

Connectivity inequalities (2) can be separated exactly in polynomial time with the following well-known algorithm. For each edge f such that $x_f^* > 0$, compute the minimum cut in graph G^+ separating edge f from the depot. If the weight of this cut is less than $2x_f^*$, then the corresponding inequality (2) is violated.

Although polynomial, the above exact algorithm is quite time consuming and usually produces a large number of violated inequalities that are very similar to each other. Therefore, we use the following labeling strategy to reduce both the number of minimum cuts computed and the number of inequalities added to the LP.

Initially, all the edges are labeled as "unexplored". An edge such as e is selected at random and the minimum cut $\delta(S)$ $(1 \in V \backslash S)$ between the depot and e is computed. Then the edge f with $x_f^* = \max \left\{ x_e^* : e \in E(S) \right\}$ is selected and the associated connectivity inequality is checked for violation. If it is not violated, the procedure starts again with another unexplored edge e'. Otherwise the violated inequality is added to the LP, and edges that are within distance d of f are labeled as "explored". The distance between two edges is defined as the minimum number of edges needed to join them. At each node of the branch-and-cut tree, d is set to 4, and it is incremented by one unit after this heuristic algorithm has been executed if it fails to find more than 4 violated inequalities, while it is decreased by one unit otherwise (but never taking a value lower than 4). Moreover, for any unexplored edge $f' \in E(S)$ such that $x_{f'}^* \geq 0.9x_f^*$, we add the associated inequality (2) to the LP if it is violated, and all the edges within distance d are labeled as explored. Then the next unexplored edge e is selected and the procedure starts again.

Note that if there are no violated inequalities, the above algorithm is equivalent to an exact one. However, if there are violated inequalities, it may not find all of them. Thus, although it is strictly speaking a heuristic algorithm, we will refer to it as the exact connectivity separation procedure.

Two more heuristic algorithms have been implemented. The first one is based on the computation of the connected components of the graph induced by the edges such that $x_e^* + y_e^* > \epsilon_1$, where ϵ_1 is a given parameter. The inequality (2) associated with each connected component and with the edge f in it having maximum x_f^* is checked for violation.

The second heuristic algorithm is based on the computation of minimum cuts on a smaller graph. First, the connected components of the graph induced by those edges e with $x_e^* \geq 1 - \epsilon_2$ are computed and are shrunk into a single vertex each, where ϵ_2 is another parameter. Then all the minimum cuts between the vertex corresponding to the component containing the depot and the other ones are computed, and the associated connectivity inequalities are checked.

5.1.2 Separation of parity inequalities

Parity inequalities (6) can be separated in polynomial time. Note that if we change $x - y$ for x in

$$(x - y)(\delta(S)\backslash F) \geq (x - y)(F) - |F| + 1, \quad \forall S \subset V, \quad \forall F \subset \delta(S) \text{ with } |F| \text{ odd},$$

we obtain the cocircuit inequalities presented in [13], which can be separated exactly in polynomial time with the algorithm described by Letchford et al. [15].

For parity inequalities with $S = \{v\}$, an exact and simple procedure (see [13]) can be applied to obtain the set of edges that define the set F in the maximally violated inequality associated with cut-set $\delta(v)$, if there is one.

A heuristic algorithm based on the computation of cut-sets in the graph G^- is also used. These cut-sets are obtained from the connected components induced by the edges with $x_e - y_e \geq \epsilon$ in G^-, where ϵ is a given parameter. For each cut-set obtained, the corresponding set F is found by applying the above procedure proposed in [13].

5.1.3 Separation of K-C inequalities

It is not known whether the problem of separating K-C inequalities can be solved in polynomial time or not, but our guess is that this problem is NP-hard. Here we propose a heuristic algorithm that is an adaptation of the one proposed in [7] for the General Routing Problem (GRP). In the GRP there is a set of edges that have to be traversed by the solution, called required edges. These edges play an important role in the definition of the partition of V in $\{M_0, \ldots, M_K\}$. In order to apply our algorithm, we will consider as required edges those with $x_e^* \geq 1 - \epsilon_1$, where ϵ_1 is a given parameter. Once we have partitioned V, we choose the set F as the set of required edges with $y_e^* \leq \epsilon_2$, where ϵ_2 is another parameter.

5.1.4 Separation of p-connectivity inequalities

As with the K-C inequalities, we do not know whether the separation problem of p-connectivity inequalities is NP-hard or not. We have devised a heuristic algorithm that seems to work well. The input for this algorithm is a cut-set $(S : V\backslash S)$ for which the corresponding connectivity inequality (2) is tight. Let us suppose that $1 \in V \backslash S$ and choose $S_0 = V \backslash S$. In order to determine the remaining sets $S_1, \ldots, S_p$, we compute the connected components C_i in the subgraph induced by the edges $e \in E(S)$ with $x_e^* \geq 1 - \epsilon$ in $G(S)$, where ϵ is a given parameter. For each pair C_i, C_j of such components, we compute

$$s_{ij} = 2x^*(V_i : V_j) - 2 \min \left\{ x^*_{e_i}, x^*_{e_j} \right\},$$

where V_i and V_j are the sets of vertices in components C_i and C_j, and e_i and e_j are the edges in C_i and C_j with the highest value of x_e^*. Note that s_{ij} represents the saving obtained in the left-hand side of the p-connectivity inequality

$$(x + y)(\delta(S_0)) + 2 \sum_{1 \leq r < t \leq p} x(S_r : S_t) - 2 \sum_{r=1}^{p} x_{e_r} \geq 0,$$

after shrinking components C_i and C_j. We shrink the components with maximal s_{ij}. This process is repeated while a positive saving s_{ij} is achieved. The sets obtained with this procedure define sets $S_1, \ldots, S_p$ and the corresponding p-connectivity inequality is checked for violation.

5.2 The cutting-plane algorithm

At each iteration of the cutting plane algorithm the separation procedures are called in the following order:

1. The first heuristic algorithm for separating connectivity inequalities is applied with $\epsilon_1 = 0$. If no violated inequalities are found, it is called again with $\epsilon_1 = 0.25, 0.5$. If this fails, the second heuristic algorithm is run with different values of $\epsilon_2 = 0, 0.1, 0.2, 0.3, 0.4$ while it does not find any violated inequality.
2. The exact separation algorithm for connectivity inequalities is applied only if the previous heuristics fail. Since this routine is too time consuming, we have tried to reduce the number of times it is executed for large-size instances (more than 1,300 edges). When the improvement obtained in the lower bound with the last executions is insignificant, this procedure is no longer called at this point of the cutting-plane and it is relegated to step 8.
3. For each cut-set obtained by the second connectivity heuristic and the exact procedure whose associated connectivity inequality is tight, the p-connectivity heuristic separation algorithm is called.
4. Parity inequalities with $S = \{v\}$ are separated exactly for every vertex $v \in V$.
5. If the previous algorithm fails, the heuristic procedure for parity inequalities is applied with $\epsilon = 0, 0.25, 0.5$.
6. If no violated parity inequalities are found, the exact procedure is applied.
7. If no violated connectivity inequalities have been found, the heuristic algorithm for separating K-C inequalities is executed consecutively with parameters

$(\epsilon_1, \epsilon_2) \in \{(0, 0), (0, 0.2), (0.2, 0), (0.2, 0.2)\}$, while it does not find any violated inequality.

8. If no violated inequalities of any type have been found and the exact separation procedure for connectivity inequalities has not yet been applied at this iteration, this exact algorithm is now run.

The above cutting-plane procedure is applied at each vertex of the tree until no new violated inequalities are found.

6 Computational results

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, the CPLEX presolve and heuristic algorithms and cut generation are turned off, the optimality gap tolerance is set to zero, and strong branching and the depth-first search are selected. Tests were run on an Intel Core 2 2.40 GHz and 2 GB RAM. All the data instances are publicly available [9]. Since there is no previous exact algorithm for the resolution of the MBCPP, we have compared our algorithm with the one by Araoz et al. [2] for the PCARP which, as mentioned before, is a special case of the MBCPP. In order to do that, we have tested the branch-and-cut algorithm described here on a set of 118 instances with the same underlying graphs as those used in [2]. Their characteristics are shown in Table 1(a). All of them were originally Rural Postman Problem instances, in which the solution has to traverse a certain subset of edges called required edges. Benefits are generated differently for required and non-required edges. The net benefit associated with the first traversal of an edge e is randomly generated in the intervals $[0, 2c_e]$ and $[-c_e, 0]$ for required and non-required edges, respectively, while the net benefit associated with the second traversal of any edge e is given by $-c_e$. Since this strategy is the same one used in [2], although the benefits may differ, we think that the two sets of instances are similar enough to be used to compare the performance of both procedures.

In order to test our algorithm on instances of larger sizes, we have used the original RPP instances from which the WRPP instances in [8] were obtained. The characteristics of these instances are shown in Table 1(b). From these sets of RPP instances, two different sets of MBCPP instances using different strategies have been created to generate the net benefits. The net benefits of the first one have been generated using the same strategy described before. To check whether the difficulty of the instances depends on the benefits structure, we have generated the net benefits of the second set of instances completely at random in $[-c_e, c_e]$.

Table 2 reports the computational results obtained by the exact algorithm (AFM) proposed in [2] on the PCARP instances described in that paper and with our exact algorithm (CPRS) on the equivalently generated MBCPP instances. All the instances were solved to optimality by both exact procedures. Columns headed “Gap” and “# opt” show the average percentage gap for all the instances in each set and the number of optimal solutions obtained by the cutting-plane algorithm of the corresponding exact procedures, respectively. The average number of nodes in the branch-and-cut

Table 1 Instance characteristics

Set	# inst	$\|V\|$	$\|E\|$
(a)			
D16	9	16	31–32
D36	9	36	72
D64	9	64	128
D100	9	100	200
G16	9	16	24
G36	9	36	60
G64	9	64	112
G100	9	100	180
R20	5	20	37–75
R30	5	30	70–112
R40	5	40	82–203
R50	5	50	130–203
P	24	7–50	10–184
AlbaidaA	1	102	160
AlbaidaB	1	90	144
(b)			
B3	3	318–490	630–873
B4	3	357–498	884–1,114
B5	3	388–498	1,106–1,326
B6	3	409–498	1,392–1,537
C3	3	502–737	1,013–1,319
C4	3	534–746	1,339–1,693
C5	3	582–749	1,705–2,005
C6	3	622–750	2,080–2,269
D3	3	661–979	1,297–1,738
D4	3	708–996	1,867–2,182
D5	3	783–999	2,361–2,678
D6	3	817–999	2,793–3,073

tree and the average time taken by the CPRS procedure are given in the columns "Nodes" and "Time". The last two columns show the average time employed by the cutting plane and the overall cut-and-branch procedure (C&B) in [2]. All the times are given in seconds. It can be seen that our cutting plane obtains a higher number of optimal solutions, 110 compared with 94, and the average gap is a bit lower, 0.14% compared with 0.37%. Moreover, the times taken to solve all the instances optimally are considerably lower with our algorithm. We would like to point out that the times from the AFM algorithm were obtained with a much slower machine (Sun ULTRA 10 at 440 Mhz, 1 GB RAM).

Table 3 contains the average number of violated inequalities found by the different separation algorithms in the branch-and-cut procedure. Columns two to four show the connectivity inequalities added by the two heuristics and the exact separation procedures described in Section 5.2.1. The violated parity inequalities found by the

Table 2 Computational results on the instances described in Araoz et al. [2]

Set	# inst	CPRS				AFM			
		Cutting plane		B & C		Cutting plane			C & B
		Gap	# opt	Nodes	Time	Gap	# opt	Time	Time
D16	9	0.00	9	0.00	0.05	0.51	8	0.3	0.68
D36	9	0.00	9	0.00	0.10	0.12	5	14.58	29.74
D64	9	0.00	9	0.00	0.21	0.14	7	105.34	212.39
D100	9	0.00	9	0.00	1.75	0.40	4	1,890.71	3,270.28
G16	9	0.00	9	0.00	0.03	0.00	9	0.28	0.28
G36	9	0.00	9	0.00	0.12	0.00	9	18.31	18.31
G64	9	1.11	8	0.44	0.67	1.85	8	139.97	236.77
G100	9	0.45	7	1.11	3.52	0.44	6	2,798.35	3,079.73
R20	5	0.00	5	0.00	0.05	0.26	4	0.4	0.41
R30	5	0.00	5	0.00	0.11	0.00	5	2.7	2.7
R40	5	0.00	5	0.00	0.13	0.00	4	3.69	3.80
R50	5	0.06	4	0.20	0.24	0.07	4	60.86	61.21
P	24	0.07	20	0.21	0.07	0.44	20	1.97	2.69
Alb-A	1	0.00	1	0.00	0.38	0.19	0	562.5	589.7
Alb-B	1	0.00	1	0.00	0.22	0.00	1	25.15	25.15
Average	118	0.14	110	0.17	0.53	0.37	94	387.15	530.96

Table 3 Average no. of cuts added during the branch-and-cut algorithm on the instances in [2]

	Connectivity			Parity			K-C	p-conn
	H1	H2	Exact	H	Ex1v	Ex		
D16	8.4	1.2	1.6	1.4	16.7	1.6	0.2	1.1
D36	17.6	2.4	4.3	5.1	47.0	0.2	0.2	2.9
D64	24.3	8.6	14.4	15.1	83.1	3.4	2.0	7.4
D100	40.6	39.6	38.7	23.4	130.1	93.3	10.6	32.4
G16	9.0	1.9	0.2	0.2	16.0	0.3	0.0	0.0
G36	22.7	10.3	3.6	3.6	47.2	8.4	0.8	3.1
G64	37.3	20.4	17.8	7.6	80.3	24.3	4.9	34.2
G100	52.7	101.8	50.1	18.2	131.2	173.8	13.4	86.0
R20	10.4	0.8	2.4	1.6	21.8	0.4	0.0	0.0
R30	23.8	1.8	7.0	10.2	41.4	3.8	0.0	1.8
R40	26.0	2.2	3.2	4.4	45.0	14.2	0.8	0.0
R50	33.2	6.6	11.6	9.8	66.2	20.6	0.0	2.2
P	7.8	1.3	2.7	1.6	25.0	2.5	0.9	1.3
Alb-A	43.0	6.0	14.0	8.0	93.0	44.0	0.0	0.0
Alb-B	33.0	11.0	6.0	4.0	70.0	10.0	0.0	2.0
Average	22.4	15.1	11.7	7.2	55.9	25.9	2.7	11.2

Table 4 Computational results on the large size instances

Set	# inst	Net benefits as in [2]					Random net benefits				
		Gap	# opt 0	# opt	Nodes	Time	Gap	# opt 0	# opt	Nodes	Time
B3	3	0.18	0	3	3.00	55.2	0.02	1	3	3.33	190.9
B4	3	0.08	1	3	4.33	119.8	0.01	2	3	1.33	100.8
B5	3	0.00	3	3	0.00	63.4	0.00	3	3	0.00	70.3
B6	3	0.00	3	3	0.00	42.3	0.00	3	3	0.00	58.6
C3	3	0.03	1	3	1.33	251.4	0.01	1	2	0.67	857.9
C4	3	0.00	2	3	0.33	503.0	0.00	3	3	0.00	420.3
C5	3	0.01	1	3	2.67	831.4	0.00	3	3	0.00	385.2
C6	3	0.00	3	3	0.00	410.0	0.00	3	3	0.00	317.4
D3	3	0.01	1	3	0.67	1,265.6	0.00	1	2	1.00	1,896.2
D4	3	0.00	2	3	0.33	2,366.3	0.00	3	3	0.00	3,578.8
D5	3	0.00	3	3	0.00	1,252.8	0.00	3	3	0.00	350.9
D6	3	0.00	3	3	0.00	380.1	0.00	3	3	0.00	660.4
Average	36	0.06	23	36	1.06	628.5	0.01	29	34	0.53	703.2

heuristic and the exact method for $|S| = 1$ and the general case are shown in columns five to seven, respectively. Finally, columns eight and nine report the average number of violated K-C and p-connectivity inequalities found for each set of instances. It can be seen that violated parity inequalities are found more frequently than the other types of inequalities, followed by connectivity inequalities. However, note that they are the only two types of inequalities for which an exact separation algorithm is known. Moreover, the separation procedure for K-C inequalities is only executed when all the connectivity inequalities are satisfied.

The results obtained with our algorithm on the large-size instances are shown in Table 4. The column headed "# opt 0" shows the number of instances optimally solved by the cutting-plane algorithm, while the number of optimal solutions found by the branch-and-cut algorithm for each set is given in the column "# opt". Furthermore, the column "Gap" reports the average percentage gap at the root node for those instances for which an optimal solution has been found, i.e. the unsolved instances have not been taken into account. As far as the instances with benefits generated as in [2] are concerned, note that 23 out of 36 instances were optimally solved at the root node with the cutting-plane algorithm. For the other 13 instances, the branch-and-cut algorithm needed a very small number of nodes to solve them to optimality. This behavior is probably due to the low average gaps obtained at the root node. All but one of the instances were solved to optimality in <1 h of CPU time. The remaining instance, belonging to set D4a, took 1 h and 14 min. Regarding the instances with random benefits, 33 out of 36 instances were solved to optimality in <1 h of CPU time. As for the other three, one of them was solved in 69 min while the other two could not be solved in 2 h. Note that the number of instances optimally solved by the cutting-plane algorithm is even higher in this case, 29 out of 36. The gaps obtained, as well as the number

Table 5 Average no. of cuts added on the large-size instances with benefits generated as in [2]

	Connectivity			Parity			K-C	p-conn
	H1	H2	Exact	H	Ex1v	Ex		
B3a	200.7	492.0	373.3	211.0	465.0	300.7	12.7	110.3
B4a	74.7	63.3	559.0	291.3	525.3	702.3	48.7	83.0
B5a	50.0	22.0	154.0	269.3	490.0	84.3	33.0	9.3
B6a	25.3	4.0	24.3	158.7	449.7	44.7	5.3	1.7
C3a	251.3	995.3	1,073.7	312.0	668.0	1,186.7	1.3	179.3
C4a	120.3	51.0	535.0	369.3	736.7	397.3	16.7	53.0
C5a	65.3	8.7	609.3	357.0	724.3	984.3	16.3	19.3
C6a	37.0	0.7	105.3	364.7	727.3	812.7	10.0	6.3
D3a	447.7	2,261.0	1,599.0	549.7	993.0	4,195.0	6.3	317.3
D4a	174.7	204.3	1,123.3	523.0	979.7	1,604.3	72.7	119.0
D5a	124.7	18.0	745.7	457.3	973.3	367.0	33.7	31.3
D6a	68.0	3.7	90.3	328.3	933.3	379.3	0.0	37.7
Average	136.6	343.7	582.7	349.3	722.1	921.6	21.4	80.6

of nodes in the branch-and-cut tree are very low which, in addition to the reasonable computing times, show that our algorithm is robust and behaves excellently.

The number of violated inequalities found for the instances with net benefits as in [2] is shown in Table 5. In general, this number seems to decrease when the density of the graphs increases. The heuristic algorithms for separating connectivity and parity inequalities seem to do a good job identifying violated inequalities, which helps to reduce the number of executions of the corresponding, and considerably more time consuming, exact algorithms. Table 6 reports the same numbers for the instances with random net benefits. We notice here that the number of violated inequalities found decreases more drastically than before when the density of the underlying graphs increases. It is worth noting that the behavior of the heuristic for separating parity inequalities is quite good, and that the number of p-connectivity inequalities found is very high in the low density instances and seems to be strongly related to the number of violated connectivity inequalities identified.

7 Conclusions

In this paper, an IP formulation for the undirected MBCPP has been proposed for the first time. We have presented a study of its associated polyhedron and introduced several families of valid inequalities inducing facets of it. Some of these families are based on previously known ones for other related arc routing problems. Although previously known cocircuit inequalities can be applied to this problem, we have presented a stronger version that is facet-inducing. A new class of facet-defining inequalities (p-connectivity inequalities), that we think could be extended to other arc routing problems, has also been presented. The separation problems for all these families have

Table 6 Average no. of cuts added on the large-size instances with benefits generated at random

	Connectivity			Parity			K-C	p-conn
	H1	H2	Exact	H	Ex1v	Ex		
B3b	176.0	220.3	1,388.3	176.3	461.0	615.0	98.0	351.0
B4b	53.7	9.3	145.7	297.0	518.3	620.0	45.7	33.3
B5b	4.0	0.3	19.0	263.7	479.3	80.0	0.3	0.0
B6b	1.7	0.0	2.0	149.0	449.0	82.0	0.0	0.7
C3b	225.7	358.7	3,416.7	308.0	737.7	3,219.3	46.0	998.3
C4b	48.7	8.7	400.0	187.7	643.3	74.0	2.7	75.0
C5b	5.7	3.0	62.3	364.3	707.0	102.0	0.0	1.0
C6b	0.3	0.3	7.7	381.3	759.3	38.7	0.0	0.3
D3b	387.3	1,280.3	8,580.3	474.3	1,006.0	3,274.7	80.3	1,322.0
D4b	56.0	17.7	2,617.3	496.0	958.0	174.3	5.3	139.3
D5b	2.7	0.0	11.3	278.0	884.0	19.7	0.0	0.0
D6b	0.7	0.0	9.0	317.0	941.0	59.0	0.0	0.3
Average	80.2	158.2	1,388.3	307.7	712.0	696.6	23.2	243.4

been studied and several heuristic and exact procedures for their resolution have been proposed. Furthermore, we have presented here a branch-and-cut algorithm for the MBCPP resolution. The computational results have shown that its behavior is very good, being capable of solving to optimality instances of up to 1,000 vertices and 3,000 edges.

Acknowledgments The authors wish to thank the Ministerio de Innovación y Ciencia/FEDER of Spain (projects MTM2009-14039-C06-02, MTM2010-19576-C02-02 and DE2009-0057) and Junta de Andalucía/FEDER (grant number FQM-5849) for its support. They also thank two anonymous referees for their careful reading of the manuscript and for their many suggestions and comments that have helped to improve the contents and readability of the paper.

References

1. Aráoz, J., Fernández, E., Franquesa, C.: The clustered price-collecting arc-routing problem. Transp. Sci. **43**, 287–300 (2009)
2. Aráoz, J., Fernández, E., Meza, O.: Solving the prize-collecting rural postman problem. Eur. J. Oper. Res. **196**, 886–896 (2009)
3. Aráoz, J., Fernández, E., Zoltan, C.: Privatized rural postman problems. Comput. Oper. Res. **33**, 3432–3449 (2006)
4. Archetti, C., Feillet, D., Hertz, A., Speranza, M.G.: The undirected capacitated arc routing problem with profits. Comput. Oper. Res. **37**, 1860–1869 (2010)
5. Barahona, F., Grötschel, M.: On the cycle polytope of a binary matroid. J. Comb. Theory B **40**, 40–62 (1986)
6. Corberán, Á., Fernández, E., Franquesa, C., Sanchis, J.M.: The windy clustered prize-collecting problem. Transp. Sci. **45**, 317–334 (2011)
7. Corberán, Á., Letchford, A.N., Sanchis, J.M.: A cutting-plane algorithm for the general routing problem. Math. Progr. **90**, 291–316 (2001)

8. Corberán, Á., Plana, I., Sanchis, J.M.: A branch & cut algorithm for the windy general routing problem and special cases. Networks **49**, 245–257 (2007)
9. Corberán, Á., Plana, I., Sanchis, J.M.: Arc Routing Problems: Data Instances. http://www.uv.es/corberan/instancias.htm
10. Corberán, Á., Sanchis, J.M.: A polyhedral approach to the rural postman problem. Eur. J. Oper. Res. **79**, 95–114 (1994)
11. Feillet, D., Dejax, P., Gendreau, M.: The profitable arc tour problem: solution with a branch-and-price algorithm. Transp. Sci. **39**, 539–552 (2005)
12. Franquesa, C.: The Clustered Prize-collecting Arc Routing Problem. PhD Thesis, Technical University of Catalonia, Barcelona (2008)
13. Ghiani, G., Laporte, G.: A branch-and-cut algorithm for the undirected rural postman problem. Math. Progr. **87**, 467–481 (2000)
14. Lenstra, J.K., Rinnooy Kan, A.H.G.: On general routing problems. Networks **6**, 593–597 (1976)
15. Letchford, A.N., Reinelt, G., Theis, D.O.: Odd minimum cut-sets and b-matchings revisited. SIAM J. Discret. Math. **22**, 1480–1487 (2008)
16. Malandraki, C., Daskin, M.S.: The maximum benefit chinese postman problem and the maximum benefit traveling salesman problem. Eur. J. Oper. Res. **65**, 218–234 (1993)
17. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York (1988)
18. Orloff, C.S.: A fundamental problem in vehicle routing. Networks **4**, 35–64 (1974)
19. Pearn, W.L., Chiu, W.C.: Approximate solutions for the maximum benefit Chinese postman problem. Int. J. Syst. Sci. **36**, 815–822 (2005)
20. Pearn, W.L., Wang, K.H.: On the maximum benefit Chinese postman problem. OMEGA **31**, 269–273 (2003)
21. Reinelt, G., Theis, D.O.: Transformation of facets of the general routing problem polytope. SIAM J. Optim. **16**, 220–234 (2005)