European Journal of Operational Research 276 (2019) 826–839

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

The periodic rural postman problem with irregular services on mixed graphs

Enrique Benavent[a], Ángel Corberán[a], Demetrio Laganà[b,*], Francesca Vocaturo[c]

[a] DEIO, Universitat de València, Dr. Moliner 50, Burjassot 46100, Valencia, Spain
[b] DIMEG, Università della Calabria, via Pietro Bucci Cubo 41/C, Arcavacata di Rende 87036, CS, Italy
[c] DESF, Università della Calabria, via Pietro Bucci Cubo 0/C, Arcavacata di Rende 87036, CS, Italy

ARTICLE INFO

Article history:
Received 7 August 2018
Accepted 22 January 2019
Available online 30 January 2019

Keywords:
Routing
Periodic Arc Routing Problem
Mixed Rural Postman Problem
Branch-and-Cut

ABSTRACT

In this paper, we deal with an extension of the rural postman problem in which some links of a mixed graph must be traversed a given number of times over a time horizon. These links represent entities that must be serviced a specified number of times in some subsets of days (or periods) of the time horizon. The aim is to design a set of minimum-cost tours, one for each day/period of the time horizon, that satisfy the service requirements. We refer to this problem as the periodic rural postman problem with irregular services (*PRPP–IS*). Some practical applications of the problem can be found in road maintenance operations and road network surveillance, for example. In order to solve the *PRPP–IS*, we propose a mathematical model and a branch-and-cut algorithm. As far as we know, this is the first exact method devised for a periodic arc routing problem. In the solution framework, constraints ensuring connectivity and other valid inequalities are identified by using specific separation procedures. Most valid inequalities consider the particular nature of the *PRPP–IS*. We show the effectiveness of the solution approach through an extensive experimental phase.

1. Introduction

The periodic routing problem consists of designing vehicle routes for all the days of a given time horizon, or planning period, in order to meet specific service requirements. The problem was first introduced by Beltrami and Bodin (1974) in an article about garbage collection. The wide applicability of the problem has led to a large number of scientific articles dealing with theoretical concepts, formulation aspects, and solution methods. In detail, the entities requiring a service, named *required elements*, correspond to all (or to some subsets of) the components of the graph representing the street network, i.e. vertices, arcs and edges. Generally, a required element does not need a service every day, but must be serviced at least once (or a specified number of times) over the time horizon. The usual objective is to minimize the total distance traveled by the vehicles over the planning period. Note that the decision maker may plan no route on a specific day for reasons of convenience if it is compatible with the service requirements.

Periodic routing problems are traditionally classified in two classes: Periodic vehicle (or node) routing problem (*PVRP*), where the required elements are located on the vertices of a graph, and periodic arc routing problem (*PARP*), where the required elements are located along the arcs/edges.

PVRPs have been studied extensively. We refer the reader to the article of Campbell and Wilson (2014) for a recent survey on these problems. According to Campbell and Wilson, a first classification of the *PVRPs* can be carried out in terms of how customers' requests are satisfied, i.e., by picking up products (recyclable materials, infective waste, etc.), by delivering them (hospital linens, groceries, etc.), or by performing on-site services (e.g., maintenance of elevators and escalators). Campbell and Wilson also distinguish the scientific articles in how the allowable schedules are defined and, consequently, in the number of feasible alternatives that can be generated (e.g., some authors enforce constraints on the minimum and maximum spacing between consecutive services, whereas other ones specify that the service days must be equally spaced over the time horizon). The objective function is another element of differentiation in Campbell and Wilson (2014). Most articles on *PVRPs* present the objective of minimizing total route cost or some related metric. Anyway, other different objectives exist in this research context (e.g., fleet size minimization and profit maximization objectives).

* Corresponding author.
E-mail addresses: enrique.benavent@uv.es (E. Benavent), angel.corberan@uv.es (Á. Corberán), demetrio.lagana@unical.it, demetriosalvatore.lagana@unical.it (D. Laganà), vocaturo@unical.it (F. Vocaturo).

https://doi.org/10.1016/j.ejor.2019.01.056
0377-2217/

The scientific literature proposes many variants and extensions of the *PVRP*. Among these, the *PVRP* with service choice generalizes the definition of periodicity. More specifically, in the *PVRP* with service choice, introduced by Francis, Smilowitz, and Tzur (2006), the delivery frequency for the required elements is defined by the mathematical model, i.e., it is a choice of the decision maker. The authors assume that each required element asks for a minimum number of visits, but is willing to accept a higher number of visits. The required element and/or the system benefits from more frequent visits, and the event is accounted for in the objective function (in fact, generally more frequent visits correspond to better services). Thus, the *PVRP* with service choice exploits possible efficiencies from combined routing and service decisions. Francis and Smilowitz (2006) present a continuous approximation model for the same problem.

Although the applications for *PARPs* are equally interesting, the scientific literature on the topic is still scarce. Ghiani, Musmanno, Paletta, and Triki (2005) deal with the periodic rural postman problem (*PRPP*) on an undirected graph. In this problem, every required edge must be serviced a given number of times over an m-day planning period in such a way that service days are equally spaced. The *PRPP* contains the rural postman problem (*RPP*) as a special case. For this NP-hard problem, the authors propose a heuristic algorithm. Chu, Labadi, and Prins (2004, 2005, 2006) and Lacomme, Prins, and Ramdane-Chérif (2005) refer to the multi-period version of the capacitated arc routing problem, named periodic capacitated arc routing problem (*PCARP*). Although the problem is the same, different conditions and graphs are considered in the various articles. For instance, in Chu, Labadi, and Prins (2005) the *PCARP* is defined on an undirected graph where each required edge has to be serviced a given number of times in the time horizon. In addition, it has a specific set of allowed day combinations. A combination is a set of possible service days (the number of days defining the combination is equal to the number of services required in the time horizon). Finally, the demand to be serviced for any combination and any day belonging to it is known. The authors report a simple example in order to better explain the representation. In waste collection, each street (required edge) has a daily production and the time horizon, corresponding to a week, can be considered cyclic (thus, the demand for the first day of the combination is equal to the sum of the daily productions since the last service in the previous horizon). For instance, for a street with a daily production equal to 100 kg of waste, and 2 services per week, a possible combination is (Tuesday, Friday) with demands for these days respectively equal to 400 kg and 300 kg. The aim of the problem in Chu et al. (2005) is to determine a day combination for each required edge and a set of vehicle routes for each day that minimizes the total cost. The following constraints are considered: (*i*) each required edge is visited according to its requirement over the planning horizon, but at most once in each day; (*ii*) each route starts and ends at the depot; (*iii*) a required edge assigned to a day is serviced by a single route in that day; and (*iv*) the vehicle capacity is respected. For this problem, the authors propose an integer linear program, two insertion methods and a two-phase heuristic algorithm. Computational results show that the insertion methods are very fast but the two-phase algorithm yields better solutions. Chu, Labadi, and Prins (2006) propose a greedy heuristic algorithm and a scatter search method for the *PCARP*. Their computational experience shows that the scatter search method clearly outperforms the greedy heuristic algorithm. In Chu et al. (2004) the same authors refer to a different model where the sets of possible service periods are not known. In this model, a minimum and maximum spacing between two successive visits for a required edge are considered. Lacomme et al. (2005) describe several versions encountered in practice and consider more realistic networks (mixed multigraphs, prohibited turns and turn penalties, etc.). The authors present a memetic algorithm based on a sophisticated crossover operator. Monroy, Amaya, and Langevin (2013) introduce the *PCARP* with irregular services. In order to explain the concept of “irregularity”, the authors refer to graphs where some arcs must be serviced twice during the first five days of the week and once during the weekend. In addition, the days for service may vary from one week to the other. This type of irregular service is common in many rural or coastal areas of Spain, where the population, and therefore the amount of waste to collect, strongly varies along the months of the year (Gómez, Pacheco, & Gonzalo-Orden, 2015). Specifically, in the *PCARP* with irregular services the required elements must be serviced a number of times in given subsets of days (or periods) of the time horizon. The objective is to maximize the services taking into account the weight (importance) of the arcs. The authors propose a mathematical programming formulation and a heuristic solution approach that includes two phases: assignment step and routing step. They present computational experiments on a new set of instances including the results of the heuristic, an upper bound based on a relaxation of the mathematical formulation and optimal solutions for a group of small instances.

The problem we address is an arc routing problem (Corberán & Laporte, 2014; Mourão & Pinto, 2017) that belongs to the *PARP* class with irregular services. Specifically, we introduce the *PRPP* with irregular services (*PRPP–IS*) defined on a mixed graph. The problem consists of designing a set of minimum-cost vehicle routes, one for each period of the time horizon, satisfying the service requirements. Since the *PRPP–IS* is a natural extension of the *RPP*, it may arise in classic arc routing contexts as garbage collection or street cleaning, for example. However, we mainly refer to road maintenance operations and road network surveillance. In this context, the activities are carried out periodically, and different roads may have different service requirements (number of visits/traversals/passages) according to their importance in the network. Marzolf, Trépanier, and Langevin (2006) report a practical example of road network monitoring by referring to the Quebec Ministry of Transportation. The road monitoring is conducted daily by patrol trucks with the aim of maintaining the safety and the viability of the road network by a fast detection of the various incidents occurring on it. This activity requires a strong interaction with other services (e.g., road signs, markings, road works) and the routes for monitoring are planned for two weeks in advance according to a route hierarchy. In this typology of activities, the number of visits also depends on the time, as evidenced by Monroy et al. (2013). For instance, some roads may require more visits during the weekend because there is more traffic and less during the working week (and vice versa for other roads). In order to consider this feature, a subdivision of the time horizon in subsets of different size is needed. The number of visits varies depending on these subsets.

The transformation of routing problems into equivalent ones defines a solution approach quite often used in the scientific literature. Transformations of routing problems with required elements corresponding to arcs and edges (i.e., arc and general routing problems) into routing problems with required elements corresponding to vertices (i.e., vehicle routing problems) were described by Tagmouti, Gendreau, and Potvin (2007) and Ciancio, Laganà, and Vocaturo (2018), among others. Huang and Lin (2014) provided an example of transformation of a *PARP* with refill points. We do not trace back our problem to an equivalent one since the transformation sensibly would increase the size of the instances to be solved.

In this article, contributions are made both in directly modeling a *PARP*, defined on a mixed graph as the most general setting, and in proposing an exact algorithm for its solution. In particular, we present a branch-and-cut algorithm for the *PRPP–IS*. As far as we know, this is the first exact method devised for a periodic arc routing problem. There are many successful applications of

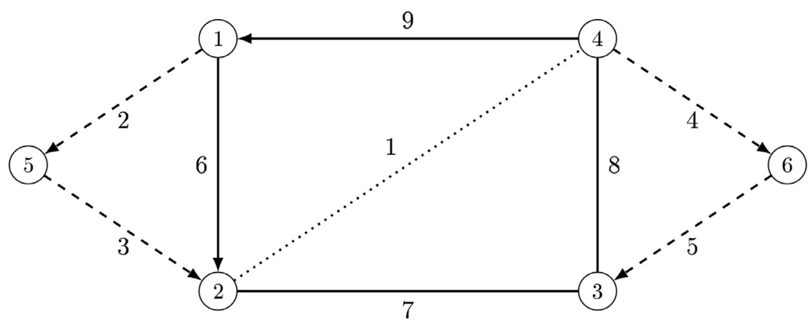


Fig. 1. A simple problem instance.

branch-and-cut methods to routing problems (see Archetti, Bertazzi, Laganà, & Vocaturo, 2017; Archetti, Corberán, Plana, Sanchis, & Speranza, 2016; Belenguer et al., 2016; Bosco, Laganà, Musmanno, & Vocaturo, 2013; Corberán, Plana, Rodríguez-Chía, & Sanchis, 2013; Irnich, Laganà, Schlebusch, & Vocaturo, 2015 for some recent references). The main component of our algorithm is a cutting-plane procedure that identifies violated inequalities of several classes, most of which consider the specific nature of the *PRPP–IS*.

The remainder of the paper is organized as follows. In Section 2, a formal description of the *PRPP–IS* is given. In Section 3, we propose an integer programming formulation for the problem, while more valid inequalities for the *PRPP–IS* are presented in Section 4. The branch-and-cut algorithm used to solve the problem is described in depth in Section 5 and the computational results are presented in Section 6. Finally, some conclusions are drawn in Section 7.

2. Problem description

Let $G = (V, E, A)$ be a strongly connected mixed graph, where $V = \{1, \dots, n\}$ is the set of vertices, including the depot (vertex 1), E is the set of edges, and A is the set of arcs. Let $E_R \subseteq E$ and $A_R \subseteq A$ be the sets of required edges and arcs, respectively. We use $L_R = A_R \cup E_R$ to represent the set of required links (i.e., arcs and edges). There is a non-negative cost $c_l = c_{ij}$ associated with the traversal of each link $l = (i, j)$ of the graph. Moreover, a service cost $c_l^s = c_{ij}^s$, $c_{ij}^s \geq c_{ij}$, is associated with each required link $l = (i, j)$. We consider a finite and discrete time horizon $H = \{1, \dots, |H|\}$ of $|H|$ periods or days. We call SP to any subset of periods/days of H. Each required link $l \in L_R$ has associated a set P_l of disjoint SPs. Each SP $T \in P_l$ has associated a *frequency* $f_T^l \leq |T|$ that indicates the number of times that link l must be serviced along the days of T. We assume that a required link must be serviced at least once over the time horizon, but it cannot be serviced more than once in the same day.

In practical applications, it is possible that for some required links l there are some days not included in any $T \in P_l$. In this case, and without loss of generality, we add a new SP containing these days and associate a zero frequency to it. Consequently, we assume that, for each required link $l \in L_R$, set of SPs P_l forms a partition of H. The problem consists of finding a set of routes, one for each day, minimizing the total cost, while satisfying the service requirements of each link. Let K be the set of all route indices. Since a route index corresponds to a day (and vice versa), K and H coincide. Therefore, in the following, we use K to define both the set of the route indices and the time horizon. From this perspective, an SP also denotes a subset of route indices.

We report a simple example in order to better explain the notation. Consider the strongly connected mixed graph represented in Fig. 1, where the numbers close to the links are the link identifiers. We suppose that all links are required. Service and traversing costs of each link are not represented in the figure as they are not relevant for the purpose of the example. The time horizon includes seven days (a week). In addition, consider the following SPs:

- $T_1 = \{Monday, Tuesday\}$,
- $T_2 = \{Wednesday, Thursday\}$,
- $T_3 = \{Friday, Saturday, Sunday\}$,
- $T_4 = \{Monday, Tuesday, Wednesday, Thursday\}$,
- $T_5 = \{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday\}$.

Link $l = 1$, corresponding to edge (2,4), is represented by a dotted line and is associated with $P_l = \{T_1, T_2, T_3\}$, links $l = 2$, $l = 3$, $l = 4$, and $l = 5$ are represented by dashed lines and are associated with $P_l = \{T_3, T_4\}$, while links $l = 6$, $l = 7$, $l = 8$ and $l = 9$ are represented by solid lines and are associated with $P_l = \{T_5\}$. We assume that all frequencies are equal to 1. Consequently, link 1 must be serviced once over the first SP (i.e., Monday and Tuesday), once over the second SP (i.e., Wednesday and Thursday), once over the third SP (i.e., Friday, Saturday, and Sunday), and so on for the other links. Since all links are required, we can refer to the problem of the example as the periodic Chinese postman problem with irregular services, which is a special case of the *PRPP–IS*.

3. Mathematical formulation

Given S, $S' \subset V$, $(S\colon S')$ denotes the set of links with an endpoint in S and the other in S', $E(S\colon S')$ the set of edges between S and S', and $A(S\colon S')$ the set of arcs from S to S'. Hence, $\left(S : S'\right) = E\left(S : S'\right) \cup A\left(S : S'\right) \cup A\left(S' : S\right)$. Furthermore, $\delta(S) = (S : V\backslash S)$, $A^+(S) = A(S : V\backslash S)$, $A^-(S) = A(V\backslash S : S)$, $A(S) = A^+(S) \cup A^-(S)$, and $E(S) = E(S : V\backslash S)$ are called *cutsets*. In addition, $\gamma(S)$ represents the set of links with both endpoints in S. These sets are denoted in a similar way when restricted to the required links only: $A_R^+(S)$, $\delta_R(S)$, $(S\colon S')_R$, etc. For the sake of brevity, we represent singleton $\{i\}$ by simply i. We can formulate the *PRPP–IS* by defining the following decision variables.

- For each arc $a = (i, j) \in A$ and each route $k \in K$, let x_{ij}^k be the number of times that arc $a = (i, j)$ is traversed in route k.
- For each edge $e = (i, j) \in E$ and each route $k \in K$, let x_{ij}^k and x_{ji}^k be the number of times that edge $e = (i, j)$ is traversed in route k from i to j and from j to i, respectively.
- For each required link $l = (i, j) \in L_R$ and each route $k \in K$, let $y_l^k = y_{ij}^k$ be a binary variable that takes value 1 if link $l = (i, j)$ is serviced in route k, and value 0 otherwise.

The corresponding model is:

$$Min \quad \sum_{k \in K} \sum_{l \in L_R} (c_l^s - c_l) y_l^k + \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} \left(x_{ij}^k + x_{ji}^k\right) + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \tag{1}$$

$$s.t.: \sum_{(i,j)\in A^+(i)} x_{ij}^k + \sum_{j:(i,j)\in E(i)} x_{ij}^k$$
$$= \sum_{(j,i)\in A^-(i)} x_{ji}^k + \sum_{j:(i,j)\in E(i)} x_{ji}^k \quad \forall\, i \in V, \forall\, k \in K, \tag{2}$$

$$\sum_{(i,j)\in A^-(S)} x_{ij}^k + \sum_{(i,j)\in E: i\in V\setminus S, j\in S} x_{ij}^k \geq y_l^k \quad \forall\, S \subseteq V\setminus\{1\},$$
$$\forall\, l \in \gamma_R(S), \forall\, k \in K \tag{3}$$

$$x_{ij}^k \geq y_{ij}^k \quad \forall\, (i,j) \in A_R, \forall\, k \in K \tag{4}$$

$$x_{ij}^k + x_{ji}^k \geq y_{ij}^k \quad \forall\, (i,j) \in E_R, \forall\, k \in K \tag{5}$$

$$\sum_{k\in T} y_l^k = f_T^l \quad \forall\, l \in L_R, \forall\, T \in P_l \tag{6}$$

$$y_l^k \in \{0,1\} \quad \forall\, l \in L_R, \forall\, k \in K \tag{7}$$

$$x_{ij}^k \geq 0 \text{ and integer} \quad \forall\, (i,j) \in A, \forall\, k \in K \tag{8}$$

$$x_{ij}^k, x_{ji}^k \geq 0 \text{ and integer} \quad \forall\, (i,j) \in E, \forall\, k \in K. \tag{9}$$

Objective function (1) minimizes the total cost. Eq. (2) represent *flow constraints*; for each route, they model the symmetry conditions at each vertex. Inequalities (3), called *connectivity constraints*, impose that the required links are serviced by routes that are connected and connected to the depot. Since they are defined for each subset of vertices, connectivity inequalities are exponential in number. Inequalities (4) and (5) force each vehicle to traverse the required links it serves (arcs and edges, respectively). Eq. (6), called *frequency constraints*, ensure that the service requirements of the required links are satisfied in each SP. Note that the required frequency service would be satisfied with equality in the optimal solution even in the case of representation through inequality since the traversing cost of any link is not greater than its servicing cost. Thus, if a link is serviced by the set of routes of an SP a number of times greater than the required frequency, we may substitute the remaining services of the link by just traversals with no increase in the cost. Finally, constraints (7)–(9) define variable domains. Note that, together with constraints (8) and (9), the flow constraints also imply parity conditions at each vertex. In addition, note that quantity $\sum_{k\in K}\sum_{l\in L_R}(c_l^s - c_l)y_l^k$ in (1) represents the total additional cost for serving all links over the time horizon. Since the total number of services in a feasible solution is fixed through Eq. (6), this quantity is constant and therefore not relevant for the routing decisions.

The algorithm proposed in the following for the solution of the *PRPP–IS* is based on the mathematical formulation just described. Other formulations concerning *PARPs* were introduced in the scientific literature, but they were not used to solve instances of realistic size. In fact, only small instances were solved by using these formulations with the aim of performing a comparison with the results provided by a heuristic algorithm. Furthermore, our formulation is different from the other formulations concerning *PARPs* for several aspects. For instance, we do not encode the day combinations as proposed in Chu et al. (2005) or do not have a maximization objective as proposed in Monroy et al. (2013).

4. Valid inequalities

To effectively solve the *PRPP–IS* with a branch-and-cut algorithm, the linear relaxation of (1)–(9) should be strengthened. To do this, some valid inequalities are proposed in this section. In what follows, and in order to simplify the exposition, we introduce some further notation. For any subset of links $L \subseteq A\cup E$, $x^k(L)$ denotes $\sum_{(i,j)\in L\cap A} x_{ij}^k + \sum_{(i,j)\in L\cap E}(x_{ij}^k + x_{ji}^k)$. For a subset $L \subseteq L_R$, $y^k(L)$ denotes $\sum_{l\in L} y_l^k$. In addition, given $L \subseteq L_R$ and $T \in P_l$, $f(L,T) = \sum_{l\in L: T\in P_l} f_T^l$ denotes the sum of the service frequencies of all the links in L that must be serviced in T.

In general, we call *aggregate inequalities* those involving the variables corresponding to all (or some) routes, and *disaggregate inequalities* those involving the variables corresponding to a single route. If the sum involves a subset of routes, then the name of the inequalities includes the aggregation field (e.g., *SP aggregate* refers to the routes associated with an SP).

4.1. Parity inequalities

In this Section we introduce several families of valid inequalities related to the parity condition of the vertices in any feasible solution for the *PRPP–IS*.

4.1.1. Aggregate parity inequalities

Given a vertex subset $S \subset V$ with $|\delta_R(S)|$ odd, we have the well known *aggregate parity inequalities* (or *R-odd inequalities*) for the *PRPP–IS*:

$$\sum_{k\in K} x^k(\delta(S)) \geq |\delta_R(S)| + 1, \quad \forall\, S \subset V \quad \text{such that } |\delta_R(S)| \text{ is odd.} \tag{10}$$

They are based on the fact that each required link has to be serviced at least once in the time horizon. For this reason, if the cardinality of cutset $\delta_R(S)$ is odd, it has to be crossed at least $|\delta_R(S)| + 1$ times.

Note that inequalities (10) are dominated by constraints (2), (4), and (6) whenever $\delta_R(S)$ only includes arcs. Since each arc $(i, j) \in \delta_R(S)$ has to be serviced at least once in the time horizon, constraints (4) and (6) ensure that $\sum_{k\in K} x_{ij}^k \geq 1$. Therefore, we have $\sum_{k\in K} x^k(\delta_R(S)) \geq |\delta_R(S)|$. Obviously, the inequality remains valid if it is extended to all links of the cutset: $\sum_{k\in K} x^k(\delta(S)) \geq |\delta_R(S)|$. We recall that constraints (2) ensure symmetry conditions for each vertex and, consequently, for each subset of vertices. If $\delta_R(S)$ is odd and only includes arcs, then the flow constraints guarantee that cutset $\delta(S)$ is crossed at least once more. On the contrary, R-odd inequalities can be violated when $\delta_R(S)$ is odd but includes some edges. Consider the case in which edge $(i, j) \in \delta_R(S)$ is traversed once in route k but in different directions in presence of solutions with fractional values for the decision variables. Condition $x_{ij}^k = x_{ji}^k = 0.5$ could lead to violations of inequality (10) that include edge (i, j) with coefficient 1 in its right-hand side, without violations of the symmetry conditions.

4.1.2. SP aggregate parity inequalities

Although valid, *R*-odd inequalities (10) may not be really binding because the required links may have to be serviced several times along the time horizon. We have strengthened them by referring only to the routes in an SP instead of considering all the routes in the time horizon. In particular, given $T \in P_l$ and $S \subset V$ with $f(\delta_R(S), T)$ odd, we introduce the *SP aggregate parity inequalities*:

$$\sum_{k\in T} x^k(\delta(S)) \geq f(\delta_R(S), T) + 1 \quad \forall\, S \subset V \text{ and } \forall\, T \quad \text{such that}$$
$$f(\delta_R(S), T) \text{ is odd.} \tag{11}$$

Proposition 1. *SP aggregate parity inequalities (11) are valid for the PRPP–IS.*

Proof. Note that $f(\delta_R(S), T)$ represents the number of times that the links in $\delta_R(S)$ must be serviced by the set of routes corresponding to T. Then, if this number is odd, cutset $\delta(S)$ has to be crossed at least once more by this set of routes. □

Inequalities (11) are also dominated by constraints (2), (4), and (6) whenever $\delta_R(S)$ includes arcs only. Since each arc

$l=(i,j)\in\delta_R(S)$ has to be serviced exactly f_T^l times in T, and constraints (4) and (6) ensure that $\sum_{k\in T} x_{ij}^k \geq f_T^l$, we have $\Sigma_{k\in T}x^k(\delta_R(S))\geq f(\delta_R(S), T)$ and, hence, $\Sigma_{k\in T}x^k(\delta(S))\geq f(\delta_R(S), T)$. If $\delta_R(S)$ includes arcs only and $f(\delta_R(S), T)$ is odd, then flow constraints (2) guarantee that cutset $\delta(S)$ is crossed at least once more in T.

4.1.3. Disaggregate parity inequalities

There are also parity inequalities that involve a single route. In particular, given a vertex subset $S\subset V$ and a subset $F\subseteq\delta_R(S)$ with $|F|$ odd, we call *disaggregate parity inequality* associated with route k to the following one:

$$x^k(\delta(S)) \geq 2y^k(F) - |F| + 1. \tag{12}$$

They are also known as *cocircuit inequalities* (see Barahona & Grötschel (1986)) and have also been used in other arc routing problems (see, for example, Belenguer and Benavent (1998) and Ghiani and Laporte (2000)). The validity of these inequalities for the *PRPP–IS* can be proved as follows. Since $|F|$ is odd, if vehicle k serves all links in F, then it has to traverse the cutset at least $|F|+1$ times. In this case, $y^k(F)=|F|$ and the right-hand side of the inequality is equal to $|F|+1$. Therefore, $x^k(\delta(S))\geq|F|+1$ satisfies the condition. If vehicle k serves all links in F except one, then $y^k(F)=|F|-1$ and the right-hand side of the inequality is $|F|-1$. In this case, the inequality is trivially satisfied. The other cases are trivial too.

4.1.4. P-aggregate parity inequalities

Inequalities (12) can be generalized from a single route to several routes. Given a subset of P routes $\bar{K}=\{k_1,k_2,\ldots,k_P\}$, with $P<|K|$, a vertex subset $S\subset V$, and a subset $F\subseteq\delta_R(S)$, let us denote $\bar{f}(F,\bar{K})=\sum_{l\in F}\sum_{T\in P_l}\min(f_T^l,|T\cap\bar{K}|)$. If $\bar{f}(F,\bar{K})$ is odd, we call *P-aggregate parity inequality* the following one:

$$\sum_{k\in\bar{K}} x^k(\delta(S)) \geq 2\sum_{k\in\bar{K}} y^k(F) - \bar{f}(F,\bar{K}) + 1. \tag{13}$$

Proposition 2. *P-aggregate parity inequalities (13) are valid for the PRPP–IS for any subset of routes $\bar{K}$ and any subset of required links $F\subseteq\delta_R(S)$ if $\bar{f}(F,\bar{K})$ is odd.*

Proof. For each required link l in F, the maximum value that $\sum_{k\in\bar{K}} y_l^k$ can take is $\sum_{T\in P_l}\min(f_T^l,|T\cap\bar{K}|)$. In particular, for each $T\in P_l$, if $T\cap\bar{K}=\emptyset$, i.e. no day/route of $T\in P_l$ belongs to $\bar{K}$, then $\min(f_T^l,|T\cap\bar{K}|)=0$. On the contrary, if $T\cap\bar{K}=T$, i.e. each day/route of $T\in P_l$ belongs to $\bar{K}$, then $\min(f_T^l,|T\cap\bar{K}|)=f_T^l$. Whenever $T\cap\bar{K}\neq\emptyset$ and $T\cap\bar{K}\neq T$, the value of $\min(f_T^l,|T\cap\bar{K}|)$ depends on the case. If we consider all links in F, then we obtain $\sum_{k\in\bar{K}} y^k(F)\leq\bar{f}(F,\bar{K})$. Then, $\bar{f}(F,\bar{K})$ represents, for the days defined by $\bar{K}$, the maximum number of services for the links belonging to F, according to their service plan. Whenever the maximum number of services is reached, i.e., $\sum_{k\in\bar{K}} y^k(F)=\bar{f}(F,\bar{K})$, the right-hand side of inequality (13) is equal to $\bar{f}(F,\bar{K})+1$. In this case, given that $\bar{f}(F,\bar{K})$ is odd, inequality (13) holds since the routes in $\bar{K}$ have to jointly traverse the cutset at least $\bar{f}(F,\bar{K})+1$ times. □

It is important to note that inequalities (13) are not dominated by the sum of the disaggregate parity inequalities associated with single routes $k_1,k_2,\ldots,k_P$. However, as the following proposition proves, the P-aggregate inequality reduces to the SP aggregate inequality associated with T when $\bar{K}=T$ and $F=\delta_R(S)$.

Proposition 3. *If $F=\delta_R(S)$ and $\bar{K}$ corresponds to T, then inequalities (13) and (11) coincide.*

Proof. We focus on a specific SP, denoted as T, by setting $\bar{K}=T$ in (13). When $F=\delta_R(S)$, we have $\bar{f}(F,\bar{K})=\sum_{l\in\delta_R(S):T\in P_l}\min(f_T^l,|T|)=\sum_{l\in\delta_R(S):T\in P_l} f_T^l=f(\delta_R(S),T)$. Note that $\bar{f}(F,\bar{K})$ odd in this case means $f(\delta_R(S), T)$ odd. In addition, according with constraints (6), $\Sigma_{k\in T}y^k(\delta_R(S))$ is equal to the total number of services for links in $\delta_R(S)$ for T, i.e. $\sum_{k\in T} y^k(\delta_R(S))=\sum_{l\in\delta_R(S):T\in P_l} f_T^l=f(\delta_R(S),T)$. Consequently, inequality (13) becomes: $\sum_{k\in T} x^k(\delta(S))\geq f(\delta_R(S),T)+1$. □

4.2. K-C inequalities

We present here several families of valid inequalities that consider simultaneously the connectivity and parity conditions on vertices and subsets of vertices.

4.2.1. Aggregate and strengthened aggregate K-C inequalities

Aggregate K-C inequalities (associated with all the routes) were initially proposed for the min-max K-vehicles windy *RPP* by Benavent, Corberán, Plana, and Sanchis (2011). They are based on the K-C inequalities proposed in Corberán and Sanchis (1994) for the undirected *RPP* and avoid infeasible solutions as the one depicted in Fig. 2a. In this figure, required edges are represented in solid lines and non-required ones in dashed lines. Clearly, the "solution" represented in the figure is infeasible because, since vertices 2 and 3 are odd, it does not correspond to an Eulerian tour. The K-C inequality violated by this "solution" (see Corberán & Sanchis (1994)) is

$$(Q-2)(x_{12}+x_{23})+x_{24}+x_{47}+x_{37}\geq 2(Q-1)+(Q-2)*\alpha,$$

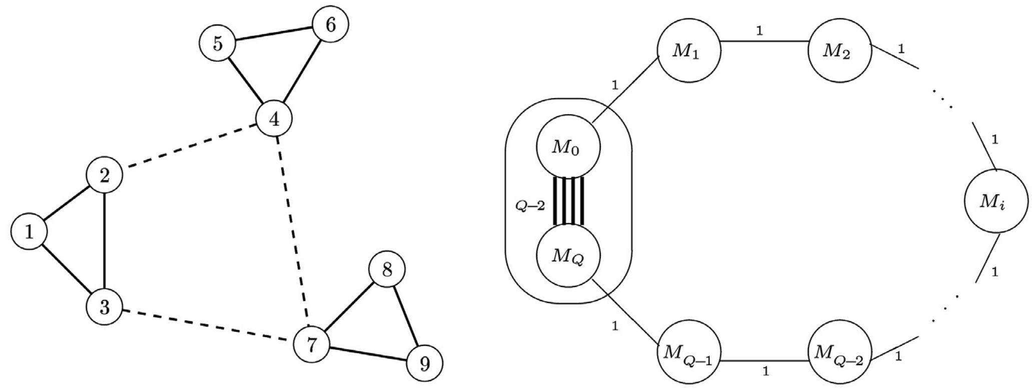
where $Q=3$ is the number of connected components induced by the required edges and $\alpha=2$ is the number of required edges between node 2 and nodes 1 and 3. Note that the left-hand side of the above inequality is 5 while the right-hand side is 6. It can be seen that 6 is the minimum cost (with the described coefficients) for a tour connecting the three connected components in an "even way" (for instance, doubling edges (2,4) and (4,7) and removing edge (3,7), or by adding another copy of edge (2,3) to the edges in the figure). Note also that if Fig. 2a contained a fourth connected component, the minimum cost for such a tour would be 8, which forces the coefficients of the edges between nodes 2 and 1 and 3 to be 2 (=$Q-2$).

Aggregate K-C inequalities are defined on an underlying configuration (see Fig. 2b) consisting of a partition of V into $Q+1$ sets, $\{M_0,M_1,\ldots,M_{Q-1}, M_Q\}$, such that each subset of vertices V_i associated with the connected components induced by the required links (called R-set) is contained in one of the vertex sets $M_0\cup M_Q, M_1,\ldots,M_{Q-1}$, each M_i contains at least an R-set, the induced subgraphs $G(M_i)$ are connected, and $(M_0{:}\ M_Q)$ contains a positive and even number of required links. Note that each subset M_i is represented as a single node in the figure, and all the (non-required) links between vertices belonging to M_i and M_{i+1} are represented by the edge in the figure joining these two nodes. The required links in $(M_0{:}\ M_Q)$, $(M_0{:}\ M_Q)_R$, are represented in bold. The coefficients a_{ij} of the variables in inequality (14) are shown in Fig. 2b, where each number represents the coefficient of the variable associated with the traversal of the link. The coefficients associated with variables corresponding to links joining vertices of the same subset M_i are zero, and the one of a variable associated with a link not shown in Fig. 2b between sets M_i and M_j can be calculated as the cost of the shortest path from M_i to M_j using the coefficients shown in this figure.

The corresponding *aggregate K-C inequality* becomes

$$\sum_{k\in K} F(x^k) \geq 2(Q-1)+(Q-2)|(M_0{:}M_Q)_R|, \tag{14}$$

by setting $F(x^k)=\sum_{(i,j)\in A} a_{ij}x_{ij}^k+\sum_{(i,j)\in E}(a_{ij}x_{ij}^k+a_{ji}x_{ji}^k)$. A proof similar to that in Benavent et al. (2011) would prove that inequality (14) is valid for the *PRPP–IS*. Basically, it is based on the fact



(a) Solution violating a K-C inequality

(b) K-C configuration

Fig. 2. K-C inequalities.

that each set M_i should be visited and the resulting graph must be a connected and even graph.

The above inequality can be strengthened by considering the frequency associated with the links in $(M_0\colon M_Q)_R$. Let $\bar{f}_{0Q} = \sum_{l\in(M_0:M_Q)_R}\sum_{T\in P_l} f_T^l$. If $\bar{f}_{0Q}$ is even, then the *strengthened aggregate K-C inequalities* for the *PRPP–IS* are as follows:

$$\sum_{k\in K} F(x^k) \geq 2(Q-1) + (Q-2)\bar{f}_{0Q}. \tag{15}$$

Proposition 4. *Inequalities (15) are valid for the PRPP–IS if $\bar{f}_{0Q}$ is an even number.*

Proof. It is sufficient to observe that the total number of traversals of the required links between M_0 and M_Q is equal to $\bar{f}_{0Q}$ for the *PRPP–IS*. □

Note that, since $\bar{f}_{0Q} \geq |(M_0\!:\!M_Q)_R|$, inequalities (15) dominate inequalities (14) when $\bar{f}_{0Q}$ is even, while inequalities (14) are trivially satisfied if $\bar{f}_{0Q} \geq |(M_0\!:\!M_Q)_R| + 1$.

4.2.2. Disaggregate K-C inequalities

Disaggregate K-C inequalities are K-C inequalities associated with a single route. They were also introduced in Benavent et al. (2011) for the min-max K-vehicles windy *RPP*. Its main difference with aggregate K-C inequalities is that, while each required link has to be serviced at least once (and, therefore, traversed) by the set of all routes, when we consider a single route it is not known if that route will service that link or it will be serviced in a different route. Therefore, it is not mandatory for a route k to visit subsets M_i defined above for the aggregate K-C inequalities. Subsets M_i have to be visited when $y_l^k = 1$ for a link $l = (v, w)$, with $v, w \in M_i$. In what follows we present the version of these inequalities for the *PRPP–IS*.

Let $\{M_0, M_1, M_2, \ldots, M_{Q-1}, M_Q\}$ be a partition of V where each subgraph $G(M_i)$ is connected. We assume that there is a required link subset $D \subseteq (M_0\colon M_Q)_R$ such that $|D|$ is positive and even and another required subset $Z = \{l_1, l_2, \ldots, l_{Q-1}\} \subset L_R$ such that each $l_i \in \gamma_R(M_i)$. If $1 \in M_0 \cup M_Q$, for each vehicle k we can write the following *disaggregate K-C inequality*:

$$F(x^k) \geq 2y^k(Z) + (Q-2)\big(2y^k(D) - |D|\big). \tag{16}$$

We can prove that disaggregate K-C inequalities (16) are valid for the *PRPP–IS* by using arguments similar to those presented in Benavent et al. (2011). Analogously, if $1 \in M_i$ with $i \notin \{0, Q\}$, then each vehicle k is forced to visit set M_i although it does not serve link l_i. In this case, we define $Z = \{l_1, l_2, \ldots, l_{Q-1}\} \setminus \{l_i\} \subset L_R$ and the following disaggregate K-C inequality is valid for the *PRPP–IS*:

$$F(x^k) \geq 2 + 2y^k(Z) + (Q-2)\big(2y^k(D) - |D|\big). \tag{17}$$

4.2.3. SP aggregate K-C inequalities

As with parity inequalities, we can consider aggregate K-C inequalities related to a subset of routes. Among them, the most interesting are those associated with the routes of an SP because they take into account the frequency of service of the links and are specific for the *PRPP–IS*. These inequalities are described in what follows. Assume again that $\{M_0, M_1, M_2, \ldots, M_{Q-1}, M_Q\}$ is a partition of V, $D \subseteq (M_0\colon M_Q)_R$, and $Z = \{l_1, l_2, \ldots, l_{Q-1}\} \subset L_R$, with $l_i \in \gamma_R(M_i)$, such that all the links in $D \cup Z$ have to be serviced at least once in $T \in P_l$, i.e. $f_T^l > 0$, for all $l \in D \cup Z$. Consider $l_{i^*} \in \gamma_R(M_{i^*})$ such that $f_T^{l_{i^*}} = max\{f_T^{l_j} : l_j \in Z\}$ and assume that the depot is located in $M_0 \cup M_Q$.

If $f_T^{l_{i^*}} = 1$, then the *SP aggregate K-C inequality* associated with T is

$$\sum_{k\in T} F(x^k) \geq 2(Q-1) + (Q-2)f(D, T), \tag{18}$$

where $f(D, T)$ $(=\sum_{l\in D:T\in P_l} f_T^l)$ has to be an even number. When $f_T^{l_{i^*}} \geq 2$, the corresponding *SP aggregate K-C inequality* associated with T is

$$\sum_{k\in T} F(x^k) \geq 2(Q-1) + 2 + 2(f_T^{l_{i^*}} - 2)d(M_0 \cup M_Q, M_{i^*}) + (Q-2)f(D, T), \tag{19}$$

where $d(M_0 \cup M_Q, M_{i^*})$ represents the shortest distance from $M_0 \cup M_Q$ to M_{i^*} in the K-C configuration graph (i.e., $d(M_0 \cup M_Q, M_{i^*}) = \min\{i^*, Q - i^*\}$), and $f(D, T)$ has to be an even number.

If the depot is located in M_r, then $r \notin \{0, Q\}$, we choose link $l_{i^*} \in \gamma(M_{i^*})$, $i^* \notin \{0, Q, r\}$, as the one satisfying $f_T^{l_{i^*}} = max\{f_T^{l_j} : l_j \in \gamma(M_j), j \neq r\}$. If $f_T^{l_{i^*}} = 1$, then the K-C inequalities associated with T are also inequalities (18). Otherwise, when $f_T^{l_{i^*}} \geq 2$, the K-C inequalities have the following expression:

$$\sum_{k\in T} F(x^k) \geq 2(Q-1) + 2 + 2(f_T^{l_{i^*}} - 2)d(M_r, M_{i^*}) + (Q-2)f(D, T), \tag{20}$$

where $d(M_r, M_{i^*})$ is the shortest distance from M_r to M_{i^*} in the K-C configuration graph, $d(M_r, M_{i^*}) = \min\{|r - i^*|, r + Q - i^*, i^* + Q - r\}$, and $f(D, T)$ has to be an even number.

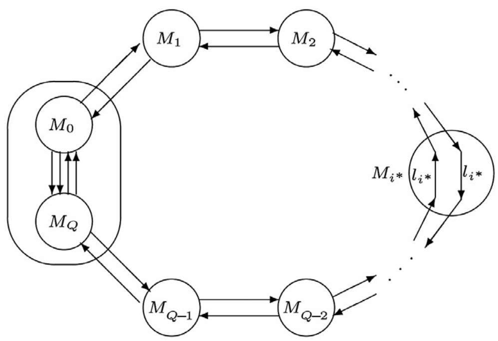


Fig. 3. Two sets of routes visiting twice M_{i^*}.

Proposition 5. *SP aggregate K-C inequalities (18)–(20) are valid for the PRPP–IS if f(D, T) is an even number.*

Proof. Let us assume that $1 \in (M_0 \cup M_Q)$. The proof in the other case would be similar. Since the routes in T service all the links in D, $\sum_{k\in T} y^k(D) = \sum_{l\in D: T\in P_l} f_T^l = f(D,T)$ and its cost is $(Q-2)f(D,T)$. Consider now the case where $f_T^{l_{i^*}} = 1$. Since the routes service all the links in Z, they have to visit sets $M_1, M_2, \dots, M_{Q-1}$ at least once. On the other hand, note that the minimum value for the left-hand side of (18) is obtained when a single route, starting and finishing at depot $1 \in (M_0 \cup M_Q)$, traverses once all the links in $D \cup Z$, and that there are only two types of routes satisfying (18) with equality:

1. routes that use path $\{M_0, M_1, \dots, M_Q\}$ once (either from M_0 to M_Q or viceversa), and traverse $f(D,T)+1$ times the links in D (remember that $f(D, T)$ is an even number), and
2. routes that use path $\{M_0, M_1, \dots, M_Q\}$ twice, once in each direction, except for one pair $(M_{i^*}, M_{i^*+1}), i \in 0, 1, \dots, Q-1$, and traverse $f(D, T)$ times the links in D.

All other routes, or sets of routes associated with T, satisfy $\sum_{k\in T} F(x^k) \geq 2(Q-1) + (Q-2)f(D,T)$, and therefore inequalities (18) hold. Consider now that $f_T^{l_{i^*}} = 2$. In this case, the minimum value for the left-hand side of (19) is $2(Q-1) + 2 + (Q-2)f(D,T)$, and is obtained when two routes, starting and finishing at the depot traverse once all the links in D, once all the links in $Z \setminus \{l_{i^*}\}$, and twice l_{i^*}. The sets of routes represented in Fig. 3 correspond to the only two types of solutions satisfying (19) with equality. It is easy to see that all other sets of routes associated with T satisfy $\sum_{k\in T} F(x^k) \geq 2(Q-1) + 2 + (Q-2)f(D,T)$, and therefore inequalities (19) hold.

Finally, if $f_T^{l_{i^*}} \geq 3$, the routes in T have to visit M_{i^*} $f_T^{l_{i^*}}$ times, all the remaining sets M_j at least once, traverse the links in D at least once, and in such a way that all the subsets $\{M_0, M_1, \dots, M_Q\}$ are incident with an even number of links. If, for example, $f_T^{l_{i^*}} = 3$, a set of routes doing this with minimal cost is the one represented in Fig. 4. Note that the minimal cost depends on the value of $f_T^{l_{i^*}} - 2$ and on the position of subset M_{i^*} and its distance to M_0 and M_Q. All other sets of routes have a cost non less than $2(Q-1) + 2 + 2(f_T^{l_{i^*}} - 2)d(M_0 \cup M_Q, M_{i^*}) + (Q-2)f(D,T)$. Hence, inequalities (19) hold. □

4.3. Symmetry-breaking inequalities

Although the routes for each day have, in principle, different requirements, there may be situations where some subsets of routes can have the same requirements and, therefore, the routes in the same subset can be interchangeable, leading to different but equivalent solutions. To illustrate this consider the example described in Section 2. The routes associated with Monday and Tuesday appear always together in any SP and, hence, the frequency requirement of every link equally concerns the Monday and Tuesday routes. Therefore, in any feasible solution we can interchange the routes of these two days obtaining an alternative solution that is in fact identical in practice. The same happens for the subsets of routes {*Wednesday*, *Thursday*} and {*Friday*, *Saturday*, *Sunday*}. The existence of this kind of alternative solutions may lead to higher computing times.

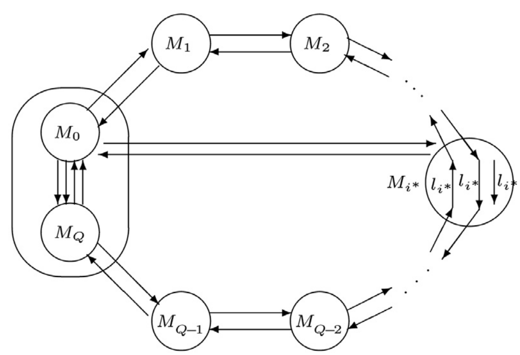


Fig. 4. A set of three routes visiting M_{i^*}.

In order to avoid this kind of symmetries we propose the following set of inequalities. Let B be an SP such that, for any $l \in L_R$ and any $T \in P_l$, either $B \subseteq T$ or $B \cap T = \emptyset$. For such a subset $B = \{k_1, \dots, k_{|B|}\}$, let $\{l_1, \dots, l_m\}$ be any ordering of the links $l \in L_R$ such that $B \in P_l$. Basically, the idea is to force the numbering of vehicles to follow the numbering of the smallest index link they service, which can be done as follows:

$$y_{l_1}^{k_1} = 1, \tag{21}$$

$$y_{l_i}^{k_j} \leq \sum_{r=1}^{\alpha_i} y_{l_r}^{k_{j-1}}, \ j = 3, \dots, |B|, i = 2, \dots, m, \tag{22}$$

where $\alpha_i = i - 1$ if $f_B^{l_i} = 1$ and $\alpha_i = i$ if $f_B^{l_i} \geq 2$.

Then, vehicle k_1 services link l_1. The second set of constraints say that if a required link l_i is serviced by vehicle k_j, then at least a previous link (or the same link if its frequency is greater than 1) has to be serviced by vehicle k_{j-1}. Links are sorted in non-increasing order of their distances to the depot and to the previous links.

5. A branch-and-cut algorithm

In this section, we describe a branch-and-cut algorithm for the *PRPP–IS* in which linear problems (LPs) are iteratively solved. In general, a branch-and-cut approach can be described as follows, in accordance with Braekers, Caris, and Janssens (2014). First, a reduced version of the problem is obtained by removing all constraint families of exponential size. Second, the linear relaxation of the reduced problem is solved. The solution value represents a lower bound on the original problem. Next, the lower bound is improved by adding cuts to the formulation and solving the corresponding LP. These cuts represent valid inequalities which are currently violated. The process continues until no more violated inequalities are found. If the solution to the current model is integer, then the optimal solution to the original problem has been found. Otherwise, two subproblems are created by branching on a fractional variable and the search continues. Whenever a subproblem is infeasible or proven to be unprofitable for the search of optimal solutions, it is discarded. The various iterations can be sketched by using a particular structure called *search tree*.

5.1. Initial relaxation and branching strategy

The initial LP is obtained from formulation (1)–(9) by removing connectivity constraints (3) and integrality restrictions in constraints (7)–(9) defining variable domains. We add to this LP the symmetry-breaking inequalities. Violated connectivity constraints are identified by using the heuristic and the exact procedure described in Section 5.2.1.

For branching we use the *strong branching strategy* implemented in the ILOG CPLEX Library. The library allows to assign different priorities to the variables and, at the current iteration, branches on a fractional variable. Variables with higher priority are the first ones checked for branching. For our problem, the priority of required links is higher than the priority of the non-required ones. The required links are ordered in a non-increasing way according to their total frequencies on the time horizon. Then, a decreasing priority is assigned to the elements of this list. If the first link $(i, j) \in L_R$ is an arc, then variables $y^1_{ij}, x^1_{ij}, \dots, y^{|K|}_{ij}, x^{|K|}_{ij}$ are checked, in this order, before examining the second link in the list. Analogously, if the first link $(i, j) \in L_R$ is an edge, then variables $y^1_{ij}, x^1_{ij}, x^1_{ji}, \dots, y^{|K|}_{ij}, x^{|K|}_{ij}, x^{|K|}_{ji}$ are checked, in this order, before examining the second link. The non-required links are sorted in non-decreasing order of their traversal costs. If the first link $(i, j) \in (A \cup E) \setminus L_R$ is an arc, then variables $x^1_{ij}, \dots, x^{|K|}_{ij}$ are checked, in this order, before examining the second link. Analogously, if (i, j) is an edge, then variables $x^1_{ij}, x^1_{ji}, \dots, x^{|K|}_{ij}, x^{|K|}_{ji}$ are checked, in this order, before examining the next link.

5.2. Separation procedures

In this section, we present the separation procedures used to identify the inequalities that are violated by the current LP solution. In the following, entities over-marked with a line represent the actual values of the variables. For instance, $\bar{x}^k_{ij}$ represents the value of variable x^k_{ij} in the current LP solution. Several procedures refer to a subgraph of G induced by the "active" links in this solution with respect to a non-negative parameter θ and to a tolerance value $\epsilon = 0.00001$. Specifically, given route k, $\bar{G}^k(\theta, \epsilon)$ denotes the subgraph of G induced by arcs (i, j) such that $\bar{x}^k_{ij} > \theta + \epsilon$ and by edges (i, j) such that $\bar{x}^k_{ij} + \bar{x}^k_{ji} > \theta + \epsilon$.

5.2.1. Separation of the connectivity inequalities

For each vehicle k, connectivity inequalities (3) can be exactly separated with the polynomial algorithm briefly described in the following. First, the algorithm determines the connected components of subgraph $\bar{G}^k(0, \epsilon)$. Then, for each link $l \in L_R$ such that $\bar{y}^k_l > \epsilon$, it computes the weight of the minimum cut separating link l from the depot in the subgraph. In this way, a subset S is defined and eventual violations of inequalities (3) are determined. If the algorithm finds at least a violated inequality, then it does not examine new subsets and stops beforehand. Such an algorithm is quite time consuming, so a simple heuristic procedure is also used. This procedure works by referring to increasing values of $\theta \in \{0, 0.25, 0.5\}$ when it does not find violated inequalities with a previous value of θ. In particular, given θ, the procedure finds the connected components of subgraph $\bar{G}^k(\theta, \epsilon)$ for each vehicle k. Then, for the set of vertices of each connected component of $\bar{G}^k(\theta, \epsilon)$ not containing the depot, inequalities (3) are checked for violation.

Algorithm 1 Connectivity Separation.

set found = FALSE and $\theta = 0$;
while $\theta \le 0.50$ and found = FALSE **do**
  for each route $k \in K$ **do**
    define the connected components of $\bar{G}^k(\theta, \epsilon)$ and heuristically separate inequalities (3);
    if violation exists **then**
      set found=TRUE;
      add violated inequalities (3) to the LP;
    end if
  end for
  if found=FALSE **then**
    set $\theta = \theta + 0.25$;
  end if
end while
if found=FALSE **then**
  use the exact procedure to identify inequalities (3);
end if

Algorithm 1 provides a simple outline of the framework used to separate connectivity inequalities.

5.2.2. Separation of the parity inequalities

Although the separation problem for disaggregate parity inequalities can be solved in polynomial time, the corresponding procedure is quite time consuming and, hence, we decided here to separate them only heuristically. Given a vertex set S, we have to check if there is a subset $F \subseteq \delta_R(S)$ for which an inequality (12) is violated. Note that such type of inequality can be rewritten as

$$x^k(\delta(S) \setminus \delta_R(S)) + x^k(\delta_R(S) \setminus F) + x^k(F) - 2y^k(F) + |F| \ge 1,$$

i.e.,

$$\sum_{(i,j)\in A(S)\setminus A_R(S)} x^k_{ij} + \sum_{(i,j)\in E(S)\setminus E_R(S)} (x^k_{ij} + x^k_{ji}) + \sum_{(i,j)\in A_R(S)\setminus (F\cap A_R)} x^k_{ij}$$
$$+ \sum_{(i,j)\in E_R(S)\setminus (F\cap E_R)} (x^k_{ij} + x^k_{ji}) + \sum_{(i,j)\in F\cap A_R} (x^k_{ij} - 2y^k_{ij} + 1)$$
$$+ \sum_{(i,j)\in F\cap E_R} (x^k_{ij} + x^k_{ji} - 2y^k_{ij} + 1) \ge 1.$$

Hence, the above inequality is violated whenever its left-hand side takes a value less than 1. Therefore, we want to find a subset F for which the left-hand side gets its minimum value for the current LP solution. In order to construct such a subset F, we proceed as follows. Starting with $F = \emptyset$, we include in F a required arc $(i, j) \in A_R(S)$ if, and only if, $\bar{x}^k_{ij} \ge \bar{x}^k_{ij} - 2\bar{y}^k_{ij} + 1$, i.e., if $\bar{y}^k_{ij} \ge 0.5$. Analogous arguments can be used to include a required edge in F. A required link can be opportunely removed/added whenever F is even. The procedure starts with $S = \{v\}$, for every $v \in V$. If it does not find violation, then examines subsets of cardinality greater than 1 according to the strategy adopted by Benavent, Corberán, Plana, and Sanchis (2009). This strategy consists of constructing the graph induced by those required arcs (i, j) for which $\bar{y}_{ij}^k \geq 0.5$ and $\bar{x}_{i,j}^k - \bar{y}_{ij}^k > \epsilon$, the required edges (i, j) for which $\bar{y}_{ij}^k \geq 0.5$ and $\bar{x}_{ij}^k + \bar{x}_{ji}^k - \bar{y}_{ij}^k > \epsilon$, those non-required arcs (i, j) for which $\bar{x}_{ij}^k > \epsilon$, and the non-required edges (i, j) for which $\bar{x}_{ij}^k + \bar{x}_{ji}^k > \epsilon$. Then, the connected components of this graph determine subsets of vertices for which the above procedure is applied.

A similar procedure is used to identify P-aggregate parity inequalities for any subset of $P = 2, 3$ routes and SP aggregate parity inequalities. These procedures are called for increasing values of $\theta \in \{0, 0.25, 0.50\}$ and stop when a violation is found.

Finally, another separation heuristic procedure for SP aggregate parity inequalities has been implemented. Given an SP, T, the method works on the graph induced by arcs (i, j) such that $\sum_{k\in T} \bar{x}_{ij}^k > \epsilon$ and by edges (i, j) such that $\sum_{k\in T} (\bar{x}_{ij}^k + \bar{x}_{ji}^k) > \epsilon$. Let v be any vertex of the above graph. The method starts with an initial set $S = \{v\}$ and, at each iteration, a new vertex is added to S. Given the current set S, for every $w \notin S$ adjacent to at least one vertex in S, the SP aggregate parity inequality is checked for $S \cup \{w\}$. Then, the vertex added to S is the one that minimizes the weight of the cutset defined by $S \cup \{w\}$. The same procedure has been applied on the graph defined by the required arcs $l = (i, j)$ such that $T \in P_l$ and $\sum_{k\in T} (\bar{x}_{ij}^k - \bar{y}_{ij}^k) > \epsilon$, the required edges $l = (i, j)$ such that $T \in P_l$ and $\sum_{k\in T} (\bar{x}_{ij}^k + \bar{x}_{ji}^k - \bar{y}_{ij}^k) > \epsilon$, the non-required arcs (i, j) such that $\sum_{k\in T} \bar{x}_{ij}^k > \epsilon$ and by the non-required edges (i, j) such that $\sum_{k\in T} (\bar{x}_{ij}^k + \bar{x}_{ji}^k) > \epsilon$.

5.2.3. Separation of the K-C inequalities

Due to the frequency constraints on the required links, aggregate K-C inequalities use to be satisfied by the LP solutions and, hence, no procedure has been implemented to separate them in our branch-and-cut algorithm.

Disaggregate K-C inequalities have been separated heuristically by using an algorithm based on the ones described in Corberán, Letchford, and Sanchis (2001) and Benavent et al. (2011). Basically, the procedure works as follows.

For a given route k, we build its corresponding support graph $\bar{G}^k(0, \epsilon)$ and label the depot and the links $l \in L_R$ such that $\bar{y}_l^k \geq \eta$ as "required", where η is a given parameter.

In order to find sets $M_0, M_1, M_2, \ldots, M_Q$, we first compute the connected components $\bar{M}_i$ induced in $\bar{G}^k(0, \epsilon)$ by the elements labeled "required". A vertex is called *external* if it is connected to at least one vertex in a different connected component $\bar{M}_j$. For each $\bar{M}_i$ with at least two external vertices and connected to at least two different connected components, we look for a partition of $\bar{M}_i$ into two subsets, such that the number of links l with $\bar{y}_l^k > \epsilon$ between these subsets is even, and an external vertex u is contained in each partition. If $\bar{M}_i$ is found, then the corresponding partition subsets are candidate for M_0 and M_Q, respectively. We repeat this step for each connected component and, if a set of candidates for M_0 and M_Q is found, then this component is inserted in a list $\mathcal{L}_{K-C}$ of possible seeds for a K-C configuration. We select seed $\bar{M}_i$ such that the weight associated with the even number of "required" links between the corresponding candidates M_0 and M_Q is maximum. The weight is computed as sum of $\bar{y}_l^k$ for each "required" link l between M_0 and M_Q. The other elements of the list are ordered in a non-increasing way according to this weight.

Once candidates for M_0 and M_Q have been found, we shrink the candidates and the remaining components $\bar{M}_j$ into a single vertex each and the values of the variables associated with all the (non-required) links between vertices belonging to M_i and M_j are added to define the weight of edge (M_i, M_j). Now, we compute a maximum weight spanning tree by iteratively adding the edge with maximum weight not forming a cycle. The tree is then transformed into a path linking the candidates for M_0 and M_Q by iteratively shrinking each non-candidate vertex with degree one into its adjacent vertex. The vertices in the path define sets $M_0, M_1, M_2, \ldots, M_Q$.

Set D is defined by the links in $(M_0\text{: } M_Q)$ labeled "required". If depot $1 \in M_0 \cup M_Q$, for each set M_j, $j = 1, \ldots, Q-1$, the link $l_j \in \gamma_R(M_j)$ that maximizes $\bar{y}_{l_j}^k$ is selected. We define $Z = \{l_1, l_2, \ldots, l_{Q-1}\}$ and check the corresponding inequality (16) for violation. On the other hand, if depot $1 \in M_i$, $i \neq 0$, $i \neq Q$, we define set Z as above except for link l_i, and check the corresponding inequality (17) for violation. Several values for η have been tried. After some computational testing with different values for η, we finally decided to set $\eta = 0.8$. If no violation is found, then another K-C configuration is built by starting from the next seed extracted from list $\mathcal{L}_{K-C}$, and the separation procedure is repeated until a violation is found or the list is empty.

For the SP aggregate K-C inequalities, we use a similar procedure. In this case, for a SP, T, we construct subgraph $\bar{G}^T(0, \epsilon)$ and label the depot and the links $l \in L_R$ such that $T \in P_l$ and $\sum_{k\in T: \bar{y}_l^k \geq \eta} \bar{y}_l^k > \epsilon$ as "required". For the separation procedure, we proceed as for the disaggregate K-C inequalities.

5.3. Cutting-plane strategy

The separation algorithms described so far have been used more extensively in the root node of the search tree than in any other node with the aim of obtaining a strong lower bound for the *PRPP–IS* before branching. In effect, the availability of a strong lower bound reduces the size of the tree and speeds up the research of the optimal solution. In order to avoid the tailing off effect in the root node, whenever the lower bound increases less than 0.1% in any 25 consecutive iterations of the cutting-plane algorithm, the separation is stopped and we resort to branching. The strategy used for calling the separation procedures at the root node can be outlined as follows:

(a) separate SP aggregate parity inequalities. For each SP, the connected components heuristic is called first and, if no violation is found, the heuristics generating sequences of subsets are called;
(b) if no violated inequality is found, separate P-aggregate parity inequalities for $P \in \{2, 3\}$ and sets S consisting of a single vertex; if the procedure fails with these vertex sets, consider those associated with the connected components of the graph built in the way described in Section 5.2.2. Whenever a violation is found, this procedure stops;
(c) if no violated inequality is found, separate connectivity constraints by calling Algorithm 1;
(d) for each route k, separate the disaggregate parity inequalities associated with k. If no violated inequality is found, separate the SP aggregate K-C inequalities and, if no violation of these inequalities has been found, try to find violated disaggregate K-C inequalities.

In what refers to the other nodes of the search tree, the strategy used for calling the separation procedures changes coherently with the reduction of the impact of the valid inequalities. Specifically, the strategy is streamlined. In addition, it is called less frequently (every ten nodes). This action allows to reduce the computational burden significantly without compromising the quality of the final solution. For the selected nodes the separation strategy is as follows:

(a) separate SP aggregate parity inequalities. For each SP, the connected components heuristic is called first and, if no

Table 1
Computational results for the `pcp1-mval` dataset.

FILE	*Instance features*						*Main results*					*Other results*					
	$\|V\|$	$\|A\|$	$\|E\|$	$\|A_R\|$	$\|E_R\|$	$\|Serv\|$	LB_0	LB	UB	GAP	SEC	CON	DIP	SAP	PAP	K-C	NOD
pcp1-mval1A	24	35	20	35	20	138	542.2	542.2	550	1.42	TL	828	808	181	64	0	43684
pcp1-mval1B	24	38	13	38	13	125	692	692	692*	–	7.49	40	2	16	0	0	0
pcp1-mval1C	24	36	17	36	17	134	636	637.167	639	0.29	TL	499	351	85	25	0	70940
pcp1-mval2A	24	28	16	28	16	116	818	818	818*	–	37.70	132	4	69	19	0	0
pcp1-mval2B	24	40	12	40	12	126	868	868	868*	–	249.79	238	68	79	54	3	0
pcp1-mval2C	24	35	14	35	14	124	768.333	776	776*	–	378.20	553	176	124	26	1	1982
pcp1-mval3A	24	33	15	33	15	124	294	297	297*	–	380.43	384	74	56	7	3	2893
pcp1-mval3B	24	29	16	29	16	112	352.25	355	355*	–	107.60	244	102	155	15	3	162
pcp1-mval3C	24	25	18	25	18	107	212.167	228	228*	–	2770.01	458	258	296	54	0	43484
pcp1-mval4A	41	69	26	69	26	236	1433	1433.2	1456	1.57	TL	1458	342	141	43	0	15460
pcp1-mval4B	41	83	19	83	19	253	1474	1475.2	1484	0.59	TL	1459	234	67	19	0	16260
pcp1-mval4C	41	82	21	82	21	254	1500	1500	1509	0.60	TL	1386	571	303	87	0	10205
pcp1-mval4D	41	83	21	83	21	260	1580	1580	1588	0.50	TL	1326	353	168	42	0	14197
pcp1-mval5A	34	74	22	74	22	244	1452.5	1453	1467	0.95	TL	1027	608	112	42	0	14969
pcp1-mval5B	34	56	35	56	35	221	1379.43	1383	1407	1.71	TL	1191	1271	590	108	0	7474
pcp1-mval5C	34	81	17	81	17	248	1699	1699	1699*	–	39.84	40	4	34	3	0	0
pcp1-mval5D	34	63	29	63	29	229	1429	1434.5	1470	2.42	TL	1371	802	176	89	0	5486
pcp1-mval6A	31	47	22	47	22	171	770	770	770*	–	143.14	225	108	69	29	0	167
pcp1-mval6B	31	44	22	44	22	165	789	794	794*	–	262.61	287	160	114	34	0	380
pcp1-mval6C	31	45	23	45	23	174	773	778	778*	–	142.93	184	43	86	51	0	11
pcp1-mval7A	40	50	36	50	36	221	927	927.941	939	1.18	TL	958	656	564	129	0	24640
pcp1-mval7B	40	66	25	66	25	225	1087	1087	1087*	–	67.83	178	8	136	4	0	11
pcp1-mval7C	40	62	28	62	28	224	958	963	971	0.82	TL	670	339	250	73	0	27500
pcp1-mval8A	30	76	20	76	20	235	1426	1430	1446	1.11	TL	924	708	97	35	0	29040
pcp1-mval8B	30	64	27	64	27	231	1327	1327	1334	0.53	TL	861	1148	167	71	0	16978
pcp1-mval8C	30	55	28	55	28	208	1336	1336	1336*	–	302.00	333	356	116	49	0	158
pcp1-mval9A	50	100	32	100	32	335	1147	1147	1154	0.61	TL	937	448	126	29	0	7850
pcp1-mval9B	50	76	44	76	44	299	1055	1055	1073	1.68	TL	747	797	397	90	0	13980
pcp1-mval9C	50	83	42	83	42	315	1033	1033	1056	2.18	TL	873	649	307	53	0	2390
pcp1-mval9D	50	93	38	93	38	325	1137.33	1138	1153	1.30	TL	1042	633	244	58	0	6269
pcp1-mval10A	50	106	32	106	32	349	1612	1617	1622	0.31	TL	672	832	206	16	0	10990
pcp1-mval10B	50	101	33	101	33	334	1598	1598	1602	0.25	TL	938	408	417	42	0	4249
pcp1-mval10C	50	100	36	100	36	338	1495	1495	1511	1.06	TL	1488	1030	258	44	0	3100
pcp1-mval10D	50	87	42	87	42	326	1374	1374	1395	1.51	TL	1146	647	293	60	0	4630
average gap %										**1.07**							
maximum gap %										**2.42**							
# optima										**13**							

violation is found, the heuristics generating sequences of subsets are called.

(b) if no violation is found, for each route k separate the disaggregate parity inequalities associated with k;

(c) if no violation is found, separate connectivity constraints by calling Algorithm 1.

A limit of 5000 violated connectivity constraints is imposed at the whole search tree. Such a limit refers only to the nodes where solutions with fractional variable values are obtained, since the algorithm always checks the connectivity of any integer solution found during the search.

6. Computational results

The instances used in the computational experiments were derived from the `mval` instances proposed by Belenguer, Benavent, Lacomme, and Prins (2006) for the mixed capacitated arc routing problem. In particular, we used the graphs of the original instances and their costs. We considered $K = \{1, 2, 3, 4, 5, 6, 7\}$ (i.e., a week). Four different sets of instances were generated as follows.

The first and the second sets share with the original `mval` instances the characteristic that all links are required. The instances belonging to these groups are named `pcp-mval`, where `pcp` stands for "periodic Chinese postman". With respect to the first set of instances (`pcp1-mval`), for each required link l, we generated a set P_l of disjoint SPs. Specifically, P_l was randomly set equal to $\{T_1, T_2, T_3\}$ or $\{T_3, T_4\}$ or $\{T_5\}$, where $T_1 = \{1, 2\}$, $T_2 = \{3, 4\}$, $T_3 = \{5, 6, 7\}$, $T_4 = \{1, 2, 3, 4\}$, $T_5 = \{1, 2, 3, 4, 5, 6, 7\}$. The number of times that required link l has to be serviced at each $T_i \in P_l$ was uniformly generated in $[1, \ldots, \lceil |T_i|/2 \rceil]$, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. The second set of instances (`pcp2-mval`) was generated from the first one by only "switching off" SPs T_1 and T_2. In particular, we have modified the `pcp1-mval` instances by setting to zero the frequencies concerning SPs T_1 and T_2 for each link l with $P_l = \{T_1, T_2, T_3\}$. Hence, this type of link has to be serviced only in T_3.

In the other sets of instances not all the links are required. Therefore, the instances belonging to these sets are named `prp-mval`, from "periodic rural postman". Specifically, the third set (`prp1-mval`) was obtained from the first one by reducing set L_R. In detail, each link in the `pcp1-mval` instance remained required with probability 0.6 (with the same SPs and frequencies). The last set (`prp2-mval`) was obtained from the third one by "switching off" SPs T_1 and T_2. Every set includes 34 instances, for a total of 136.

The branch-and-cut algorithm was coded in C/C++ using the IBM CPLEX 12.6 Concert Technology with default parameters, except the MIP emphasis, the MIP variable selection strategy and the MIP probing level that are all set equal to 3. All the tests were run on an Intel(R) Xeon(R), CPU E5335, at 2.00 GHz with 8 GB RAM and a time limit of two hours (Operating System: Linux UBUNTU 14.04). In addition, all standard CPLEX cuts were activated. The computational results are shown in Tables 1–4, where the meaning of column headings follows:

Table 2
Computational results for the `pcp2-mval` dataset.

FILE	*Instance features*						*Main results*					*Other results*					
	$\|V\|$	$\|A\|$	$\|E\|$	$\|A_R\|$	$\|E_R\|$	$\|Serv\|$	LB$_0$	LB	UB	GAP	SEC	CON	DIP	SAP	PAP	K-C	NOD
pcp2-mval1A	24	35	20	35	20	108	440.5	443.75	448	0.95	TL	818	444	209	149	0	45281
pcp2-mval1B	24	38	13	38	13	83	447	447	447*	–	27.19	71	0	18	39	1	1
pcp2-mval1C	24	36	17	36	17	104	510	510	510*	–	93.39	178	45	43	94	0	0
pcp2-mval2A	24	28	16	28	16	80	652	652	652*	–	71.09	148	34	85	17	0	0
pcp2-mval2B	24	40	12	40	12	94	691	693	693*	–	218.55	254	52	46	35	3	51
pcp2-mval2C	24	35	14	35	14	98	647	647	647*	–	100.94	183	17	62	36	1	0
pcp2-mval3A	24	33	15	33	15	86	248.15	249	249*	–	200.35	234	21	19	22	0	17
pcp2-mval3B	24	29	16	29	16	78	261	263	263*	–	107.73	202	23	53	66	0	122
pcp2-mval3C	24	25	18	25	18	85	179.488	184	184*	–	625.10	343	159	296	169	6	5739
pcp2-mval4A	41	69	26	69	26	174	1099	1116.333	1123	0.59	TL	1777	310	142	64	0	18790
pcp2-mval4B	41	83	19	83	19	193	1256	1256	1261	0.40	TL	1116	114	44	38	1	12214
pcp2-mval4C	41	82	21	82	21	194	1176.667	1176.667	1181	0.37	TL	1694	325	154	47	0	13630
pcp2-mval4D	41	83	21	83	21	184	1163	1163	1165	0.17	TL	860	131	53	93	0	19703
pcp2-mval5A	34	74	22	74	22	180	1158	1158	1161	0.26	TL	941	342	56	85	0	13126
pcp2-mval5B	34	56	35	56	35	163	1062	1074.667	1089	1.32	TL	1308	689	267	231	0	5620
pcp2-mval5C	34	81	17	81	17	176	1285	1285	1285*	–	461.00	713	83	8	26	0	604
pcp2-mval5D	34	63	29	63	29	159	1019.75	1019.75	1032	1.19	TL	1326	596	158	141	0	5858
pcp2-mval6A	31	47	22	47	22	133	618	625	634	1.42	TL	743	337	215	115	0	42260
pcp2-mval6B	31	44	22	44	22	117	563	566	566*	–	262.19	252	103	11	77	2	573
pcp2-mval6C	31	45	23	45	23	132	624	629	629*	–	182.51	193	13	70	33	2	167
pcp2-mval7A	40	50	36	50	36	151	754.75	757	757*	–	1789.85	489	211	398	183	0	5531
pcp2-mval7B	40	66	25	66	25	157	867	872	872*	–	264.63	263	41	27	40	0	425
pcp2-mval7C	40	62	28	62	28	172	788.5	798	798*	–	426.28	354	75	187	82	0	551
pcp2-mval8A	30	76	20	76	20	179	1140.21	1140.667	1148	0.64	TL	1430	441	15	60	0	15810
pcp2-mval8B	30	64	27	64	27	185	1122	1125.125	1133	0.70	TL	755	798	176	72	0	20600
pcp2-mval8C	30	55	28	55	28	150	979	981	994	1.31	TL	927	396	29	159	0	37655
pcp2-mval9A	50	100	32	100	32	261	967	967	967*	–	380.22	314	84	30	33	0	71
pcp2-mval9B	50	76	44	76	44	225	842.444	847.19	854	0.80	TL	1114	514	213	222	0	3924
pcp2-mval9C	50	83	42	83	42	241	836	836	842	0.71	TL	1162	332	123	202	0	4635
pcp2-mval9D	50	93	38	93	38	241	834.037	837.571	855	2.04	TL	1638	519	229	114	0	5070
pcp2-mval10A	50	106	32	106	32	251	1243	1243.111	1249	0.47	TL	1704	319	77	55	0	3241
pcp2-mval10B	50	101	33	101	33	248	1193	1194.5	1223	2.33	TL	1361	195	106	130	0	8533
pcp2-mval10C	50	100	36	100	36	252	1095.5	1096.5	1111	1.31	TL	1595	527	235	142	0	2390
pcp2-mval10D	50	87	42	87	42	254	1131	1131.333	1144	1.11	TL	1537	377	71	125	0	3132
average gap %										**0.95**							
maximum gap %										**2.33**							
#optima										**15**							

FILE	instance name
$\|V\|$, $\|A\|$, $\|E\|$	number of vertices, arcs, and edges, respectively
$\|A_R\|$, $\|E_R\|$	number of required arcs and edges, respectively
$\|Serv\|$	number of services
LB$_0$	lower bound at the root node
LB	best lower bound
UB	best solution value reached within the time limit (an optimal value certified by the algorithm is marked with an asterisk)
GAP	percentage optimality gap from CPLEX (it is given only for the instances not solved to optimality)
SEC	total computing time in seconds (the time limit of two hours is marked with TL)
CON	number of violated connectivity inequalities added in the cutting-plane fashion
DIP, SAP, PAP	number of violated disaggregate, SP aggregate, and *P*-aggregate parity inequalities dynamically added in the cutting-plane fashion, respectively
K-C	number of violated disaggregate and SP aggregate K-C inequalities dynamically added in the cutting-plane fashion
NOD	number of nodes in the branch-and-cut tree (apart from the root node).

The total number of optimal solutions obtained in each dataset is reported in row "# optima" of Tables 1–4. The average and maximum percentage gaps are also shown. Note that the average percentage gaps have been computed only with respect to the instances not solved to optimality.

In our opinion, the results obtained are very good, especially considering the great difficulty of this problem and the size of the benchmark instances tried, which have up to 50 vertices, 138 required links, and 349 services. As Monroy et al. (2013) point out, "it should be emphasized that the size of the search space for the *PCARP* depends on the total number of services and not on the number of required arcs". These authors also mention that Lacomme et al. (2005) consider that instances with 200 services are of very large size. Moreover, note that, in both papers, the authors are considering solving the instances by means of approximate algorithms, while we are trying to solve them to optimality.

From the tables, it can be noticed that the number of optimally solved instances is 13 for the `pcp1-mval` dataset, 15 for `pcp2-mval`, 21 for `prp1-mval`, and 25 for `prp2-mval`, for a total of 74 out of 136 instances. Among the instances solved to optimality there are four instances with more than 200 services (up to 261).

The obtained results confirm that the possibility of finding an optimal solution generally reduces with the increase of the number of services. Consequently, solving an instance of the "rural" sets is generally easier than solving the corresponding instance of the "Chinese" sets, where the number of required links and services is widely higher. In effect, the number of optimal solutions for `prp1-mval` and `prp2-mval` sets is greater than the number of optimal solutions for `pcp1-mval` and `pcp2-mval` sets. Analogously, solving an instance of datasets where some SPs were "switched off" is, in general, slightly easier than solving the original instance. In particular, the number of optimal solutions for `pcp2-mval` and `prp2-mval` sets is greater than the number

Table 3
Computational results for the `prp1-mval` dataset.

FILE	*Instance features*						*Main results*					*Other results*					
	$\|V\|$	$\|A\|$	$\|E\|$	$\|A_R\|$	$\|E_R\|$	$\|Serv\|$	LB_0	LB	UB	GAP	SEC	CON	DIP	SAP	PAP	K-C	NOD
prp1-mval1A	24	35	20	21	13	87	377	380	380*	–	207.61	143	31	112	16	4	775
prp1-mval1B	24	38	13	25	9	89	564	564	564*	–	4.81	49	0	17	5	0	0
prp1-mval1C	24	36	17	23	10	85	505	508	508*	–	198.34	206	17	42	7	0	4267
prp1-mval2A	24	28	16	14	10	64	626	634	634*	–	71.41	210	15	13	15	0	68
prp1-mval2B	24	40	12	26	7	78	639	639	639*	–	56.74	137	13	67	2	0	17
prp1-mval2C	24	35	14	22	8	77	534	535	535*	–	139.91	212	57	53	27	0	16
prp1-mval3A	24	33	15	22	8	76	237	237	237*	–	11.07	77	3	25	9	0	0
prp1-mval3B	24	29	16	18	6	59	233	233	233*	–	18.58	79	7	43	12	0	0
prp1-mval3C	24	25	18	15	11	66	182	183	183*	–	69.78	106	36	134	11	1	21
prp1-mval4A	41	69	26	41	22	152	1018.67	1021.667	1033	1.10	TL	1463	357	252	53	0	19145
prp1-mval4B	41	83	19	52	12	158	1012	1018	1018*	–	481.96	758	79	137	5	0	292
prp1-mval4C	41	82	21	50	14	158	1039	1043	1043*	–	3630.39	1310	125	196	22	0	13842
prp1-mval4D	41	83	21	48	7	135	1017	1017	1017*	–	283.44	293	27	62	21	0	28
prp1-mval5A	34	74	22	47	15	158	1099	1109	1116	0.63	TL	1660	308	238	21	0	19697
prp1-mval5B	34	56	35	39	22	149	1068	1069.929	1073	0.29	TL	1378	405	673	79	0	15094
prp1-mval5C	34	81	17	51	11	155	1263	1264.048	1269	0.39	TL	1368	213	41	12	0	33030
prp1-mval5D	34	63	29	34	19	128	966	968	968*	–	228.91	418	40	245	22	0	71
prp1-mval6A	31	47	22	30	15	112	548	554	558	0.72	TL	392	99	105	13	3	72980
prp1-mval6B	31	44	22	26	12	95	541	543	543*	–	183.88	209	7	19	9	3	236
prp1-mval6C	31	45	23	28	16	114	566	566	566*	–	188.16	276	22	44	8	0	634
prp1-mval7A	40	50	36	28	20	125	610.072	615	615*	–	268.39	221	68	262	77	0	126
prp1-mval7B	40	66	25	43	18	152	813	816	816*	–	502.40	345	46	151	42	1	1906
prp1-mval7C	40	62	28	37	19	138	746.5	755	755*	–	6995.53	479	139	205	30	0	81270
prp1-mval8A	30	76	20	45	8	128	867	867	867*	–	159.46	313	2	7	6	3	18
prp1-mval8B	30	64	27	33	16	118	817.235	825	825*	–	1789.88	622	219	114	83	0	16834
prp1-mval8C	30	55	28	35	17	133	945	959	959*	–	214.93	325	51	84	41	0	408
prp1-mval9A	50	100	32	65	20	217	832.333	836.5	841	0.54	TL	1719	432	238	41	0	10486
prp1-mval9B	50	76	44	48	27	186	753.875	753.875	769	1.97	TL	971	317	280	200	6	8120
prp1-mval9C	50	83	42	57	27	213	785	786	791	0.63	TL	1212	312	410	54	0	9097
prp1-mval9D	50	93	38	60	23	207	825.883	827.227	846	2.22	TL	1670	274	336	23	1	4340
prp1-mval10A	50	106	32	58	23	204	1049	1049	1050	0.10	TL	825	114	217	33	0	41800
prp1-mval10B	50	101	33	63	20	207	971	974.6	976	0.14	TL	1659	210	358	7	0	9949
prp1-mval10C	50	100	36	63	21	209	1008	1008	1012	0.40	TL	2429	194	201	13	0	4550
prp1-mval10D	50	87	42	58	28	219	986	986	1005	1.89	TL	1012	275	247	34	0	6551
average gap %										**0.85**							
maximum gap %										**2.22**							
# optima										**21**							

of optimal solutions for `pcp1-mval` and `prp1-mval` sets, respectively.

The trend is confirmed by the percentage gaps on the unsolved instances. Nevertheless, in this case, the performance decreases very slowly. In effect, the average percentage gaps are quite similar being equal to 1.07%, 0.95%, 0.85%, and 0.79% for the `pcp1-mval`, `pcp2-mval`, `prp1-mval`, and `prp2-mval` sets, respectively. The maximum percentage gaps are also similar. Specifically, they are equal to 2.42%, 2.33%, 2.22%, and 1.99% for the `pcp1-mval`, `pcp2-mval`, `prp1-mval`, and `prp2-mval` sets, respectively. The most positive result, in our opinion, is right represented by the percentage gap which remains below a threshold equal to 2.42% for all instances. This result provides assurances that our algorithm works well also on instances of very large size. Note also that the lower bounds obtained at the root node are very tight, which confirms that the different families of inequalities proposed in this paper provide a good description of the *PRPP–IS* polyhedron.

Tables 1–4 also show that many violated connectivity inequalities and disaggregate, SP aggregate, and P-aggregate parity inequalities have been found and added to the LPs during the execution of the cutting-plane algorithms, mainly at the root node. The number of violated disaggregate and SP aggregate K-C inequalities identified is very low, especially on the “Chinese” sets of instances. Note that the separation of these two families of inequalities is more time demanding and, hence, the corresponding procedures are called only at the root node and only when no violated inequality of any other family has been found. Moreover, violated inequalities of these families use to appear when the number of connected components induced by the services of the required links is big (for example when the number of required links and services is small). This happens in the “rural” instances, especially in the `prp2-mval` set, where they have shown to be important to close the gaps of some instances.

The contribution of the SP aggregate parity, *P*-aggregate parity and symmetry-breaking inequalities has been significant, especially the one concerning the symmetry-breaking inequalities, which try to avoid or limit the huge number of equivalent solutions, one of the most critical issues in routing problems. Since all these inequalities have been appositely designed for the *PRPP–IS*, a more accurate analysis of their contribution seems to be suitable. Table 5 summarizes their effect on the first 9 instances of sets `pcp1-mval` and `prp1-mval`. This is a group of 18 small-medium size instances, 16 of which have been solved to optimality. In particular, Table 5 reports the following information:

$\overline{\text{GAP}}$	average percentage gap
$\overline{\text{NOD}}$	average number of nodes in the branch-and-cut (apart from the root node)
$\overline{\text{SEC}}$	average computing time in seconds
OPT	number of optimal solutions

In this case the average percentage gap is computed by considering all instances (i.e., the instances solved to optimality and the unsolved instances). Rows “All”, “without SBI”, “without SBI + PAP”, and “without SBI + SAP” refer to the cases in which the whole version of the algorithm, the version without symmetry-breaking inequalities, the version without symmetry-breaking and *P*-aggregate parity inequalities, and the version without symmetry-breaking and SP aggregate parity inequalities is used, respectively.

Table 4
Computational results for the `prp2-mval` dataset.

FILE	*Instance features*						*Main results*					*Other results*					
	$\|V\|$	$\|A\|$	$\|E\|$	$\|A_R\|$	$\|E_R\|$	$\|Serv\|$	LB_0	LB	UB	GAP	SEC	CON	DIP	SAP	PAP	K-C	NOD
prp2-mval1A	24	35	20	21	13	69	315	317	317*	–	85.94	123	17	97	4	0	197
prp2-mval1B	24	38	13	25	9	55	336	349	349*	–	10.08	116	4	15	0	0	58
prp2-mval1C	24	36	17	23	10	67	416	419	419*	–	83.10	166	20	29	17	0	302
prp2-mval2A	24	28	16	14	10	46	469.667	482	482*	–	132.07	217	22	29	13	1	1139
prp2-mval2B	24	40	12	26	7	56	505	505	505*	–	160.22	253	13	54	44	13	0
prp2-mval2C	24	35	14	22	8	63	480	480	480*	–	62.03	133	14	32	14	8	0
prp2-mval3A	24	33	15	22	8	58	206	206	206*	–	56.37	125	8	8	10	1	0
prp2-mval3B	24	29	16	18	6	43	175	175	175*	–	65.78	133	25	17	13	1	0
prp2-mval3C	24	25	18	15	11	52	142	145	145*	–	126.45	175	33	84	6	3	62
prp2-mval4A	41	69	26	41	22	112	780	805	805*	–	560.65	736	73	98	29	0	587
prp2-mval4B	41	83	19	52	12	120	865.018	880	880*	–	570.59	540	41	92	13	2	906
prp2-mval4C	41	82	21	50	14	120	812.607	830	830*	–	789.98	689	37	139	72	5	620
prp2-mval4D	41	83	21	48	7	101	821	823	823*	–	6930.12	1097	80	139	18	5	37129
prp2-mval5A	34	74	22	47	15	110	842	851.556	863	1.33	TL	986	129	58	80	2	29443
prp2-mval5B	34	56	35	39	22	105	805.537	818	818*	–	3957.62	981	170	603	146	0	11993
prp2-mval5C	34	81	17	51	11	121	987	987.667	989	0.14	TL	857	145	23	16	1	66390
prp2-mval5D	34	63	29	34	19	86	681.5	694	694*	–	513.68	509	62	113	73	4	396
prp2-mval6A	31	47	22	30	15	88	481	482	482*	–	366.10	229	27	67	16	0	2980
prp2-mval6B	31	44	22	26	12	65	420	420	420*	–	620.39	289	22	17	72	2	2
prp2-mval6C	31	45	23	28	16	86	468	471	471*	–	278.23	287	32	27	41	0	1541
prp2-mval7A	40	50	36	28	20	85	505	508	508*	–	873.42	261	34	178	159	5	2125
prp2-mval7B	40	66	25	43	18	108	650	650	650*	–	275.54	177	30	0	4	2	3
prp2-mval7C	40	62	28	37	19	106	591.429	599	599*	–	2100.38	447	84	53	104	4	7294
prp2-mval8A	30	76	20	45	8	102	746	746	746*	–	186.37	274	3	4	4	4	61
prp2-mval8B	30	64	27	33	16	96	709	713	713*	–	1154.42	627	170	96	126	2	2972
prp2-mval8C	30	55	28	35	17	95	724	747	747*	–	528.32	514	76	8	124	1	2775
prp2-mval9A	50	100	32	65	20	171	712	712.5	715	0.35	TL	1045	127	64	54	3	13630
prp2-mval9B	50	76	44	48	27	140	615.333	617.5	622	0.72	TL	848	178	296	58	0	16690
prp2-mval9C	50	83	42	57	27	163	649.5	651.8	665	1.99	TL	1070	141	390	34	4	11286
prp2-mval9D	50	93	38	60	23	153	626.5	629.224	632	0.44	TL	993	178	265	114	0	13870
prp2-mval10A	50	106	32	58	23	148	866	866	867	0.12	TL	944	152	187	84	0	16450
prp2-mval10B	50	101	33	63	20	153	765.5	772	784	1.53	TL	1515	134	180	74	6	11500
prp2-mval10C	50	100	36	63	21	155	791	791	791*	–	3852.77	1087	115	358	76	1	8985
prp2-mval10D	50	87	42	58	28	173	818.6	821.821	826	0.51	TL	1315	172	135	122	2	8360
average gap %										**0.79**							
maximum gap %										**1.99**							
#optima										**25**							

Table 5
Effect of some inequalities.

	$\overline{\text{GAP}}$	$\overline{\text{NOD}}$	$\overline{\text{SEC}}$	OPT
All	0.09	9350.5	1061.64	16
Without SBI	0.20	22921.89	1836.36	15
Without SBI + PAP	0.18	23876.44	1901.35	14
Without SBI + SAP	0.18	36982	2294.73	14

For all versions, the time limit remained fixed to two hours. Whenever the time limit is reached, we consider 7200 seconds for computing $\overline{\text{SEC}}$. The table shows that removing the symmetry-breaking inequalities has a discrete impact on the number of optima (OPT). However, a more significant impact can be noticed in the values of $\overline{\text{GAP}}$, $\overline{\text{NOD}}$, and, therefore, in the computing times. Note that, in effect, the search tree increases considerably and $\overline{\text{SEC}}$ moves from 1061.64 to 1836.36 seconds. If, in addition to the symmetry-breaking inequalities, the *P*-aggregate parity inequalities or the SP aggregate parity inequalities are removed, then the results get worse in terms of number of optimal solutions and computational effort. Finally, comparing the effect of both families of inequalities, it can be noticed that SP aggregate parity inequalities seem to have a stronger effect than *P*-aggregate parity inequalities both in terms of number of nodes and computing time.

7. Conclusions

In this paper, we have studied a new periodic arc routing problem, the periodic rural postman problem with irregular services (*PRPP–IS*). For this problem, we have proposed a mathematical formulation and presented new valid inequalities, most of which consider the particular nature of the *PRPP–IS*. Based on the proposed model, we have implemented a branch-and-cut algorithm to solve the problem to optimality. The performance of this algorithm has been evaluated by carrying out computational experiments on a large set of instances derived from a well-known dataset for the mixed capacitated arc routing problem. Many instances used in the experiments are of very large size since they include more than 200 services (up to 349). The algorithm was able to optimally solve 74 out of 136 instances from scratch, i.e., without using upper bounds that would reduce the length of the tree search. Moreover, the percentage gap obtained for any instance remained below a very satisfactory threshold equal to 2.42%. As expected, the study has confirmed that the difficulty of the problem increases when the number of services increases. Nevertheless, the performance of the algorithm described in this paper decreases very slowly when the number of services increases.

As far as we know, although several periodic arc routing problems have been studied in the literature, this is the first exact algorithm proposed for such a problem. Since there are important real life applications that could be modeled and solved as a *PRPP–IS*, we plan to devise a metaheuristic capable of solving instances of even larger size and helping in the exact solution of the problem.

Acknowledgments

The work by Enrique Benavent and Ángel Corberán was supported by the Spanish Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional (FEDER) through project MTM2015-68097-P (MINECO/FEDER).

References

Archetti, C., Bertazzi, L., Laganà, D., & Vocaturo, F. (2017). The undirected capacitated general routing problem with profits. *European Journal of Operational Research, 257*(3), 822–833.

Archetti, C., Corberán, A., Plana, I., Sanchis, J. M., & Speranza, M. G. (2016). A branch-and-cut algorithm for the orienteering arc routing problem. *Computers & Operations Research, 66*, 95–104.

Barahona, F., & Grötschel, M. (1986). On the cycle polytope of a binary matroid. *Journal of Combinatorial Theory, 40*(1), 40–62.

Belenguer, J. M., & Benavent, E. (1998). The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications, 10*(2), 165–187.

Belenguer, J. M., Benavent, E., Lacomme, P., & Prins, C. (2006). Lower and upper bounds for the mixed capacitated arc routing problem. *Computers & Operations Research, 33*(12), 3363–3383.

Belenguer, J. M., Benavent, E., Martínez, A., Prins, C., Prodhon, C., & Villegas, J. G. (2016). A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. *Transportation Science, 50*(2), 735–749.

Beltrami, E. J., & Bodin, L. D. (1974). Networks and vehicle routing for municipal waste collection. *Networks, 4*(1), 65–94.

Benavent, E., Corberán, A., Plana, I., & Sanchis, J. M. (2009). Min-max k-vehicles windy rural postman problem. *Networks, 54*(4), 216–226.

Benavent, E., Corberán, A., Plana, I., & Sanchis, J. M. (2011). New facets and an enhanced branch-and-cut for the min-max *k*-vehicles windy rural postman problem. *Networks, 58*(4), 255–272.

Bosco, A., Laganà, D., Musmanno, R., & Vocaturo, F. (2013). Modeling and solving the mixed capacitated general routing problem. *Optimization Letters, 7*(7), 1451–1469.

Braekers, K., Caris, A., & Janssens, G. K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B, 67*, 166–186.

Campbell, A. M., & Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks, 63*(1), 2–15.

Chu, F., Labadi, N., & Prins, C. (2004). The periodic capacitated arc routing problem: Linear programming model, metaheuristic and lower bounds. *Journal of Systems Science and Systems Engineering, 13*(4), 423–435.

Chu, F., Labadi, N., & Prins, C. (2005). Heuristics for the periodic capacitated arc routing problem. *Journal of Intelligent Manufacturing, 16*(2), 243–251.

Chu, F., Labadi, N., & Prins, C. (2006). A scatter search for the periodic capacitated arc routing problem. *European Journal of Operational Research, 169*(2), 586–605.

Ciancio, C., Laganà, D., & Vocaturo, F. (2018). Branch-price-and-cut for the mixed capacitated general routing problem with time windows. *European Journal of Operational Research, 267*(1), 187–199.

Corberán, A., & Laporte, G. (2014). Arc routing: Problems, methods and applications. In *MOS-SIAM series on optimization*. Philadelphia: SIAM.

Corberán, A., Letchford, A. N., & Sanchis, J. M. (2001). A cutting plane algorithm for the general routing problem. *Mathematical Programming, 90*(2), 291–316.

Corberán, A., Plana, I., Rodríguez-Chía, A. M., & Sanchis, J. M. (2013). A branch-and–cut algorithm for the maximum benefit chinese postman problem. *Mathematical Programming, 141*(1–2), 21–48.

Corberán, A., & Sanchis, J. M. (1994). A polyhedral approach to the rural postman problem. *European Journal of Operational Research, 79*(1), 95–114.

Francis, P., & Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B, 40*(10), 872–884.

Francis, P., Smilowitz, K., & Tzur, M. (2006). The period vehicle routing problem with service choice. *Transportation Science, 40*(4), 439–454.

Ghiani, G., & Laporte, G. (2000). A branch-and-cut algorithm for the undirected rural postman problem. *Mathematical Programming, 87*(3), 467–481.

Ghiani, G., Musmanno, R., Paletta, G., & Triki, C. (2005). A heuristic for the periodic rural postman problem. *Computers & Operations Research, 32*(2), 219–228.

Gómez, J. R., Pacheco, J., & Gonzalo-Orden, H. (2015). A tabu search method for a bi-objective urban waste collection problem. *Computer-Aided Civil and Infrastructure Engineering, 30*(1), 36–53.

Huang, S. H., & Lin, T. H. (2014). Using ant colony optimization to solve periodic arc routing problem with refill points. *Journal of Industrial and Production Engineering, 31*(7), 441–451.

Irnich, S., Laganà, D., Schlebusch, C., & Vocaturo, F. (2015). Two-phase branch-and–cut for the mixed capacitated general routing problem. *European Journal of Operational Research, 243*(1), 17–29.

Lacomme, P., Prins, C., & Ramdane-Chérif, W. (2005). Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research, 165*(2), 535–553.

Marzolf, F., Trépanier, M., & Langevin, A. (2006). Road network monitoring: Algorithms and a case study. *Computers & Operations Research, 33*(12), 3494–3507.

Monroy, I. M., Amaya, C. A., & Langevin, A. (2013). The periodic capacitated arc routing problem with irregular services. *Discrete Applied Mathematics, 161*(4–5), 691–701.

Mourão, M. C., & Pinto, L. S. (2017). An updated annotated bibliography on arc routing problems. *Networks, 70*(3), 144–194.

Tagmouti, M., Gendreau, M., & Potvin, J. Y. (2007). Arc routing problems with time-dependent service costs. *European Journal of Operational Research, 181*(1), 30–39.