Computers & Operations Research 37 (2010) 692–699

Contents lists available at ScienceDirect

Computers & Operations Research

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

Lower bounds for the mixed capacitated arc routing problem☆

Luís Gouveia[a,c], Maria Cândida Mourão[b,c,*], Leonor Santiago Pinto[b,d]

[a] Universidade de Lisboa, Faculdade de Ciências, DEIO, Lisboa, Portugal
[b] Universidade Técnica de Lisboa, Instituto Superior de Economia e Gestão, Lisboa, Portugal
[c] Centro de Investigação Operacional, Portugal
[d] CEMAPRE – Centro de Matemática Aplicada à Previsão e Decisão Económica, Portugal

ARTICLE INFO

Available online 30 June 2009

Keywords:
Mixed capacitated arc routing
Formulations
Lower bounds

ABSTRACT

Capacitated arc routing problems (CARP) arise in distribution or collecting problems where activities are performed by vehicles, with limited capacity, and are continuously distributed along some pre-defined links of a network. The CARP is defined either as an undirected problem or as a directed problem depending on whether the required links are undirected or directed. The mixed capacitated arc routing problem (MCARP) models a more realistic scenario since it considers directed as well as undirected required links in the associated network. We present a compact flow based model for the MCARP. Due to its large number of variables and constraints, we have created an aggregated version of the original model. Although this model is no longer valid, we show that it provides the same linear programming bound than the original model. Different sets of valid inequalities are also derived. The quality of the models is tested on benchmark instances with quite promising results.

1. Introduction

Capacitated arc routing problems (CARP) arise in distribution or collecting problems where activities are performed by vehicles, with limited capacity, and are continuously distributed along some predefined links (roads, streets) of an associated network. The CARP can be either undirected or directed. In the undirected case, the required links can be served in any direction. In the directed case, the required links must be served only in the defined direction. The mixed capacitated arc routing problem (MCARP) models a more realistic scenario as it accommodates simultaneously both types of links. The MCARP is a NP-hard problem since it generalizes the CARP [19] which is known to be NP-hard.

The research on CARP lower bounding procedures, solution and modelling approaches performed in the last decade is surveyed by Wøhlk [30]. Many real world applications, such as household refuse collection, winter gritting, postal distribution, metre reading, street swiping, can be modelled either as a CARP or a MCARP. The surveys on arc routing by Assad and Golden [3], Eiselt et al. [13,14] and Dror's book [11] include many references on real world problems modelled as ARPs until the year 2000. More recent publications on arc routing real world applications include postal delivering by Irnich [20]; a real situation arising on an industrial company by Moreira et al. [25] and garbage collection, which is a main concern of municipalities (see [2,5,10,17,24,26,27]).

The MCARP study reported in this paper is motivated by a household refuse collection problem defined in a quarter of Lisbon. Old town quarters are usually represented by directed graphs, while new town quarters are defined in mixed networks.

Many CARP applications differ on the features of the system collection design, namely the number of depots and its location ([1,10,18,26], to name a few).

An approach to solve capacitated arc routing problems is by means of well known transformations into equivalent node routing problems [29,4,22,5]. The main idea is to use available and well tested methods for node routing problems. However, these transformations lead, in general, to networks that are substantially larger than the originals and many authors prefer to develop models on the original graph. This is also the approach followed on this paper.

The first formulation for the CARP was proposed by Golden and Wong [19] and includes an exponential number of constraints. It is also stated that the exponential sized set of subtour-breaking constraints may be replaced by a more compact set, based on flow variables. The lower bound provided from the LP relaxation of this formulation is known to be equal to zero (see [12]). Golden and Wong [19] did not use the compact model to get lower bounds for the CARP. Instead, a different lower bound method was developed and its bound was shown to be equal to the bound obtained from

☆ Project partially supported by FCT (POCTI-ISFL-1-152 and FEDER/POCI 2010).

* Corresponding author at: Instituto Superior de Economia e Gestão Rua Quelhas, 6, 1200-781 Lisboa, Portugal.
E-mail address: cmourao@iseg.utl.pt (M.C. Mourão).

0305-0548/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2009.06.018

the optimal value of a relaxation where capacity and connectivity constraints are omitted.

A different model for the undirected CARP was proposed, in 1998, by Belenguer and Benavent [6]. In 2003, the same authors [7] suggested a different formulation for the same problem that has only a single variable for each edge of the underlying graph, but it contains an exponential number of constraints. This formulation is shown to be non-valid, similarly to what happens with one of the models presented in this paper.

Later on, Belenguer et al. [8] developed a study on lower bounds for the MCARP based on the model defined in [7]. This non-valid model for the MCARP is similar to models presented for other mixed arc routing problems, as the mixed Chinese postman problem [28] and the mixed general routing problem [9]. The authors use this model and several valid inequalities in a cutting plane fashion to get lower bounds for the MCARP that outperformed the previous best known bounds.

In this paper, we formulate the MCARP by a compact model, and as far as we know, it is the first valid model for the MCARP given in the literature that is tested on reasonable large sized instances. We use two well known ideas to design this formulation for the problem: (i) the concept of flows to guarantee the connectivity of the solutions (see, for instance [15,16]) and (ii) the concept of indexing the variables by vehicle to guarantee a matching between trips and vehicles (see, for instance [23]). The model will be used within an ILP package to solve medium sized problems and to produce lower bounds on larger instances. Lower bounds are also obtained from the corresponding linear programming relaxation.

Our model differs from the model by Golden and Wong [19] in several aspects: (i) it formulates the mixed case while their model was developed for the undirected CARP; (ii) the flow variables have a different interpretation (here they are related with the demands to be served and in their paper flows are associated with the number of edges to serve); (iii) additional constraints are included to ensure that trips start at the depot; and (iv) extra valid inequalities are considered to strengthen the linear programming relaxation. A straightforward extended formulation of Golden and Wong [19] to the mixed CARP was tested on small instances by Lacomme et al. [21]; it also differs from ours on the above mentioned items (ii)–(iv).

Due to the vehicle indexing, the number of variables and constraints in our model is huge. Following the literature on the classic vehicle routing problem (VRP), we may try to get a more compact model, where links and flow variables are not disaggregated by vehicle. Unfortunately, in contrast with the classic VRP, it does not seem easy to find a similar valid model for the MCARP. However, we will present and discuss one such aggregated model which, although not valid, is attractive for three reasons. First, an integer optimal solution of the aggregated model is easier to compute than an optimal integer solution of the original model. An integer solution of the new model gives a lower bound on the optimal solution value of the MCARP which, as our computational experiments will show, provides competitive lower bound values for some classes of well known MCARP instances. Second, for certain instances, the integer lower bound value is equal to the value of a known heuristic solution, thus certifying the optimality of this solution. Finally, we will also prove that the linear programming relaxation values of the two models are equal. This means that the disaggregated model can be replaced by the non-valid model in order to produce the linear programming bound in a much faster way (since the aggregated model has fewer constraints and variables).

Comparing with the Belenguer and Benavent [7] formulation, the main difference between our aggregated model and their model lies on the network type (mixed versus undirected) and on the size of the models since our model is compact and theirs has an exponential number of constraints. That is, in our model capacity and connectivity constraints are enforced by using the additional flow variables and the constraints linking the two sets of variables. In [7] the authors do not use the extra set of variables but use, in turn, exponential sized sets of constraints to force connectivity.

The paper is organized as follows. In Section 2, we define the MCARP, set notation and present the two formulations, the valid and the non-valid formulation. We also prove that both formulations produce the same linear programming bound and discuss valid inequalities. Section 3 reports the results from the computational experiments on a set of known benchmark instances. The performance of the new models is compared with existent methods. A section of final remarks concludes the paper.

2. Formulations

2.1. Introduction

The terminology presented in this section reflects the fact that our study is motivated by a refuse collection problem. The problem undertaken is to plan the collection of garbage in a city with minimum total cost. The street network is described by a mixed graph. Edges characterize two way streets where zig-zag collection is allowed, i.e., the vehicle can collect the garbage in both sides of the street with a single traversal. Arcs represent either one way streets or large two way streets with no zig-zag collection. In the later case one arc for each direction should be included in the network. Nodes characterize the street crossings or dead-end streets. A special node, called *depot*, is the starting and ending point for the vehicle trips. The depot is also the dumpsite, where vehicles empty the refuse collected. A *vehicle trip* is a circuit that can be performed by a vehicle from and back to the depot while servicing the streets (network links), compatible with its capacity. The streets to be served, where there is refuse to be collected, are the *required links* or *tasks*. Some of the streets do not have refuse to be collected and they may be traversed only to ensure the connectivity of the trips. Every street (task or not) traversed by a vehicle without serving it is a *deadheading link*. For simplicity, it is assumed that each vehicle performs only one trip. Capacity and number of vehicles, demands on each street, service and deadheading costs and dump cost at depot are known.

Consider, then, the following notation:

- $\Gamma = (N, A' \cup E)$ is the mixed graph, with $A_R \subseteq A'$ and $E_R \subseteq E$ the set of required arcs and edges, respectively; and N the set of nodes, representing street crossings, dead-end streets, or the depot.
- $0 \in N$ is the depot node where every vehicle trip must start and end ($|N| = n+1$).
- $G = (N, A)$ is a directed graph where each edge from E is replaced by two opposite arcs, i.e., $A = A' \cup \{(i,j),(j,i) : (i,j) \in E\}$.
- $R \subseteq A$ is the set of required arcs in G, also named as tasks ($|R| = |A_R| + 2\,|E_R|$).
- P is the maximum number of trips allowed.
- W is the capacity of each vehicle.
- λ is the dump cost, paid every time a vehicle is emptied at the depot.
- d_{ij} is the deadheading cost of arc $(i,j) \in A$.
- c_{ij} is the service cost of arc $(i,j) \in R$.
- q_{ij} is the demand of arc $(i,j) \in R$.
- Q_T is the total demand, computed as $Q_T = \sum_{(i,j) \in A_R \cup E_R} q_{ij}$.

The problem is to find a set of no more than P vehicle trips, satisfying the vehicles capacity, starting and ending at the depot, node 0, and servicing all the tasks at minimum total cost.

In the sequel **LF** denotes the linear programming relaxation of formulation **F** and z_F^* the optimal value of **F**.

We present, next, two compact formulations for the MCARP, one that is valid (Section 2.2) and the other that is not (Section 2.3). As noted before, the non-valid formulation provides quite good lower bounds in a reasonable computation time.

2.2. Valid formulation for the MCARP

The following mixed integer linear programming is a valid model for the MCARP.

For $p=1,\dots,P$ define

- $x_{ij}^p=\begin{cases}1 & \text{if } (i,j)\in R \text{ is served by trip } p\\ 0 & \text{otherwise}\end{cases} \quad \forall (i,j)\in R$;
- y_{ij}^p is the number of times that arc $(i,j)\in A$ is deadheaded during trip p ;
- f_{ij}^p is the flow in arc $(i,j)\in A$, related with the remaining demand in trip p or in a sub-circuit in p.

(**F1**)

$$\min \quad \sum_{p=1}^{P}\left[\sum_{(i,j)\in R} c_{ij}x_{ij}^p+\sum_{(i,j)\in A} d_{ij}y_{ij}^p+\lambda\sum_{(i,0)\in A} y_{i0}^p+\lambda\sum_{(i,0)\in R} x_{i0}^p\right] \tag{1}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} y_{ij}^p+\sum_{j:(i,j)\in R} x_{ij}^p=\sum_{j:(j,i)\in A} y_{ji}^p+\sum_{j:(j,i)\in R} x_{ji}^p \quad i=0,1,\dots,n;\quad p=1,\dots,P \tag{2}$$

$$\sum_{p=1}^{P} x_{ij}^p=1 \quad \forall (i,j)\in A_R \tag{3}$$

$$\sum_{p=1}^{P}(x_{ij}^p+x_{ji}^p)=1 \quad \forall (i,j)\in E_R \tag{4}$$

$$\sum_{j:(0,j)\in A} y_{0j}^p+\sum_{j:(0,j)\in R} x_{0j}^p\le 1, \quad p=1,\dots,P \tag{5}$$

$$\sum_{j:(j,i)\in A} f_{ji}^p-\sum_{j:(i,j)\in A} f_{ij}^p=\sum_{j:(j,i)\in R} q_{ji}x_{ji}^p, \quad i=1,\dots,n;\quad p=1,\dots,P \tag{6}$$

$$\sum_{j:(0,j)\in A} f_{0j}^p=\sum_{(i,j)\in R} q_{ij}x_{ij}^p, \quad p=1,\dots,P \tag{7}$$

$$\sum_{i:(i,0)\in A} f_{i0}^p=\sum_{i:(i,0)\in R} q_{i0}x_{i0}^p, \quad p=1,\dots,P \tag{8}$$

$$f_{ij}^p\le W(y_{ij}^p+x_{ij}^p) \quad \forall (i,j)\in A,\ p=1,\dots,P \tag{9}$$

$$x_{ij}^p\in\{0,1\} \quad \forall (i,j)\in R,\ p=1,\dots,P \tag{10}$$

$$f_{ij}^p\ge 0 \quad \forall (i,j)\in A,\ p=1,\dots,P \tag{11}$$

$$y_{ij}^p\ge 0 \text{ integer} \quad \forall (i,j)\in A,\ p=1,\dots,P \tag{12}$$

The objective function, (1), represents the total cost, i.e., the service cost, the deadheading cost and the dump cost. Conditions (2) impose the continuity of trips at each node; the service of each required arc and edge is guaranteed by (3) and (4), respectively; (5) implies that the dump cost is adequately charged in the objective function; (6)–(8) are flow conservation constraints that together with the linking constraints (9) force the connectivity of the trips. Note that (6) are typical generalized flow conservation constraints on each node i, guaranteeing that if arc (j,i) is served by vehicle p, then q_{ji} units of flow are absorbed by node i. Conditions (9) impose upper bounds on the flow variables needed to guarantee the capacity constraints for each vehicle; they also imply that a flow variable is positive only if the corresponding arc is traversed by the vehicle trip thus ensuring connectivity, as stated above, with the flow conservation constraints.

In F1 the number of variables is equal to $P(2|A|+|R|)$, and the number of constraints is equal to $P(2|N|+|A|+2)+|A_R|+|E_R|$.

Note that the above formulation for the MCARP assumes that the depot cannot be used as an intermediate node. However, that situation is easily modelled if the depot is replicated, that is, an extra node, denoted by n+1, is considered with the same links as node 0. The service constraints, sets (3) and (4), for the tasks incident into node 0 must be modified accordingly.

Forbidden turn restrictions can also be included in the model, as they can easily be recognized if arcs are numbered.

The following two lemmas are needed to prove the validity of model F1 for the MCARP. As noted before, a feasible MCARP solution is a set of trips, thus satisfying the capacity of the vehicles and forming circuits from and back to the depot, which serves every required link.

Lemma 1. *The vehicle capacity requirements are satisfied by any feasible solution of* F1.

Proof. Fixing $i=0$ and adding (9) for all *arcs* $(0,j)\in A$ we obtain $\sum_{j:(0,j)\in A} f_{0j}^p\le W,\ \sum_{j:(0,j)\in A}(y_{0j}^p+x_{0j}^p)$. Using (5), the previous inequality implies $\sum_{j:(0,j)\in A} f_{0j}^p\le W$. Finally, (7) gives the required inequality $\sum_{(i,j)\in R} q_{ij}x_{ij}^p\le W$, which completes the proof. □

The next lemma shows that any feasible solution of F1 is connected.

Lemma 2. *For each trip* $p(p=1,\dots,P)$, *the graph induced by the set of arcs corresponding to variables* $x_{ij}^p=1$ and $y_{ij}^p>0$ *is connected*.

Proof. We prove this result by showing that any trip that serves tasks in a set $S\subset N\backslash\{0\}$ starts at the depot.

Assume that there is a required arc set S, served by a vehicle p, with S representing a connected component and that $0\in\bar S=N\backslash S$.

From (6) we obtain

$$\sum_{i\in S}\left(\sum_{j:(j,i)\in A} f_{ji}^p-\sum_{j:(i,j)\in A} f_{ij}^p\right)=\sum_{i\in S}\left(\sum_{j:(j,i)\in R} q_{ji}x_{ji}^p\right)=Q_S^p+Q_{\bar S,S}^p,$$

where Q_S^p is the demand served by trip p in subset S and $Q_{\bar S,S}^p$ is the demand served by trip p in arcs from $\bar S$ to S.

Defining

$$f_{X,Y}^p=\sum_{\substack{(i,j)\in(X,Y):\\ X,Y\subset N}} f_{ij}^p$$

the expression

$$\sum_{i\in S}\left(\sum_{j:(j,i)\in A} f_{ji}^p-\sum_{j:(i,j)\in A} f_{ij}^p\right)$$

can be rewritten as $f_{S,S}^p+f_{\bar S,S}^p-f_{S,S}^p-f_{S,\bar S}^p$, leading to $f_{\bar S,S}^p=f_{S,\bar S}^p+Q_S^p+Q_{\bar S S}^p$. Since $f_{S,\bar S}^p\ge 0;\ Q_{\bar S S}^p\ge 0$ and that, by assumption, $Q_S^p>0$ must hold, we obtain $f_{\bar S,S}^p>0$. Using (9) we conclude that $y_{\bar S,S}^p>0\vee x_{\bar S,S}^p>0$. Then, the trip p links $\bar S$ to S, and is linked to the depot node. □

From the two lemmas and conditions (2) we show that a feasible solution for F1 is a set of trips. Together with constraints (3) and (4) the solution also satisfies all services. Thus, it now becomes trivial to prove Proposition 1.

Proposition 1. *Formulation* F1 *is valid for the MCARP.*

Next, we present sets of valid inequalities that may improve the linear programming relaxation bound of the previous formulation and, as a consequence, may speed up the integer solver.

First, observe that

$$\sum_{p=1}^{P}\left(\sum_{j:(0,j)\in A} y_{0j}^{p} + \sum_{j:(0,j)\in R} x_{0j}^{p}\right) \geq \frac{\sum_{p=1}^{P}\sum_{j:(0,j)\in A} f_{0j}^{p}}{W}$$

by (9). Finally, by (3), (4) and (7), the last expression is equal to Q_T/W.

Since the left hand side of the previous inequality is integer, the right hand side may be rounded up to obtain

$$\sum_{p=1}^{P}\left(\sum_{j:(0,j)\in A} y_{0j}^{p} + \sum_{j:(0,j)\in R} x_{0j}^{p}\right) \geq \left\lceil \frac{Q_T}{W} \right\rceil \tag{13}$$

that corresponds to a depot degree constraint, stating that the solution must contain a minimum number of vehicles to satisfy the total demand. As will be confirmed by the computational tests (in Section 3.1), the inclusion of this condition leads, in general, to better linear programming relaxation bounds.

Constraints that impose lower bound values on the flow variables may also be stated:

$$f_{ij}^{p} \geq q_{ij} x_{ij}^{p} \quad \forall (i,j) \in R, \;\; p = 1, \ldots, P \tag{14}$$

$$f_{ij}^{p} \geq y_{ij}^{p} - 1 \quad \forall (i,j) \in A \backslash R, \;\; p = 1, \ldots, P \tag{15}$$

The first set of constraints specifies that the value of the flow on an arc served by a trip should be at least equal to its demand while the second set relates the flow in deadheading arcs with the number of times that arcs are deadheaded each trip. Again, from the computational tests reported in Section 3.1, we will see that linear programming relaxation bounds are improved when (14) and (15) are added to the model.

The existence of alternative integer solutions that differ only in their vehicle indexes, representing permutations of trips, may lead to huge computation times. The next set of constraints break some of these symmetries:

$$\sum_{j:(0,j)\in A} y_{0j}^{p} + \sum_{j:(0,j)\in R} x_{0j}^{p} \geq \sum_{j:(0,j)\in A} y_{0j}^{p+1} + \sum_{j:(0,j)\in R} x_{0j}^{p+1}, \quad p = 1, \ldots, P-1 \tag{16}$$

In a solution with p trips, these inequalities remove all the equivalent solutions with trip indexes greater than p. Again, our results give some evidence that (16) also speed up the ILP solver.

We denote by F1R the model F1 reinforced with constraints (13)–(16).

As pointed out before, the number of variables and constraints of the formulation F1 is too large. In the next section, an aggregated version of this formulation is presented and discussed.

2.3. Aggregated non-valid formulation

As explained in Section 1, in this model, variables are aggregated over the set of trips. Consider the following sets of variables:

- $x_{ij} = \begin{cases} 1 & \text{if } (i,j) \in R \text{ is served} \\ 0 & \text{otherwise} \end{cases}$, thus $x_{ij} = \sum_{p=1}^{P} x_{ij}^{p}$, $\forall (i,j) \in A_R \cup E_R$.
- y_{ij} is the number of times that arc $(i,j) \in A$ is deadheaded, thus $y_{ij} = \sum_{p=1}^{P} y_{ij}^{p}$.
- f_{ij} is the flow that traverses arc $(i,j) \in A$, thus $f_{ij} = \sum_{p=1}^{P} f_{ij}^{p}$.

(F2)

$$\min \quad \sum_{(i,j)\in R} c_{ij} x_{ij} + \sum_{(i,j)\in A} d_{ij} y_{ij} + \lambda \sum_{(i,0)\in A} y_{i0} + \lambda \sum_{(i,0)\in R} x_{i0} \tag{17}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} y_{ij} + \sum_{j:(i,j)\in R} x_{ij} = \sum_{j:(j,i)\in A} y_{ji} + \sum_{j:(j,i)\in R} x_{ji}, \quad i = 0, 1, \ldots, n \tag{18}$$

$$x_{ij} = 1 \quad \forall (i,j) \in A_R \tag{19}$$

$$x_{ij} + x_{ji} = 1 \quad \forall (i,j) \in E_R \tag{20}$$

$$\sum_{j:(0,j)\in A} y_{0j} + \sum_{j:(0,j)\in R} x_{0j} \leq P \tag{21}$$

$$\sum_{j:(j,i)\in A} f_{ji} - \sum_{j:(i,j)\in A} f_{ij} = \sum_{j:(j,i)\in R} q_{ji} x_{ji}, \quad i = 1, \ldots, n \tag{22}$$

$$\sum_{j:(0,j)\in A} f_{0j} = Q_T \tag{23}$$

$$\sum_{i:(i,0)\in A} f_{i0} = \sum_{i:(i,0)\in R} q_{i0} x_{i0} \tag{24}$$

$$f_{ij} \leq W(y_{ij} + x_{ij}) \quad \forall (i,j) \in A \tag{25}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in R \tag{26}$$

$$f_{ij} \geq 0 \quad \forall (i,j) \in A \tag{27}$$

$$y_{ij} \geq 0 \text{ integer} \quad \forall (i,j) \in A \tag{28}$$

This model is not valid for the MCARP, as the following example illustrates.

Example 1. Consider that the vehicle capacity is $W = 4$, that all required arcs have unitary demand, and the total demand is equal to 8. All the links in the underlined graph are depicted in Fig. 1. A feasible solution for F2 is given by $x_{ij} = 1$, $\forall (i,j) \in R$ (represented by thick arrows in Fig. 1); $y_{02} = y_{20} = 2$ (each represented by two dashed arrows in Fig. 1); $f_{02} = 8, f_{21} = 3, f_{14} = 2, f_{42} = 1, f_{23} = 2, f_{32} = 1, f_{26} = 3, f_{65} = 2$ and $f_{52} = 1$. All remaining variables are zero.

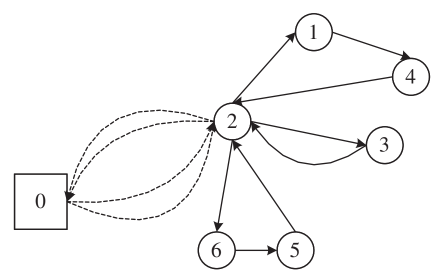

Fig. 1. Solution of the aggregated model.

For the previous solution of F2 to be feasible for the MCARP we need to be able to identify two trips without adding extra deadheading arcs, which is impossible. Note that the solution of F2 is formed by three required subcircuits, incident into node 2, with demands 3, 2 and 3; and no pair of these can fit in only one vehicle, so no two trips can be found.

The aggregated model contains only $(2|A| + |R|)$ variables and $2|N| + |A| + 2 + |A_R| + |E_R|$ constraints. In fact, if we use (19) and (20) these values can even be reduced. However, we maintain (19) and (20) to emphasize the relation with the previous formulation F1. The new model is, thus, more attractive to use within an ILP solver. In fact, our computational experiments confirm that this model, although not valid, is much easier to solve than the previous one. The results also indicate that the optimal integer solution values produced by the model are, in many cases, reasonably good lower bounds on the optimal cost of the corresponding MCARP.

As mentioned before, another interesting advantage of the aggregated model is that its linear programming relaxation value is equal to the linear programming relaxation value of the disaggregated model, F1.

Proposition 2. *Let* LF1 *be the linear programming relaxation of* F1, *with optimal value* Z^*_{LF1}, *and* LF2 *the linear programming relaxation of* F2, *with optimal value* Z^*_{LF2}. *Then* $Z^*_{LF1} = Z^*_{LF2}$.

Proof. First note that, if x^p_{ij}, y^p_{ij} and f^p_{ij} $\forall (i,j) \in A$, $p = 1, \ldots, P$ is a feasible solution of LF1, a feasible solution of LF2 with equal value can be obtained by the equalities that relate the variables from the two formulations (see the beginning of this section). To complete the proof we must show that any feasible solution of LF2 can be transformed into a feasible solution for LF1 with the same value.

Let $\overline{x_{ij}}$; $\overline{y_{ij}}$; $\overline{f_{ij}}$ be a feasible solution of LF2. Consider a solution for LF1 obtained in the following way:

$$\forall p = 1, \ldots, P; \ \ (i,j) \in A: \quad x^p_{ij} = \frac{\overline{x_{ij}}}{P}; \ y^p_{ij} = \frac{\overline{y_{ij}}}{P}; \ f^p_{ij} = \frac{\overline{f_{ij}}}{P}$$

This solution is feasible for LF1 since:

(i)

$$\forall\, p = 1, \ldots, P: \sum_{j:(i,j)\in A} (y^p_{ij} + x^p_{ij}) = \frac{1}{P} \sum_{j:(i,j)\in A} (\overline{y_{ij}} + \overline{x_{ij}})$$
$$\underset{(18)}{=} \frac{1}{P} \sum_{j:(i,j)\in A} (\overline{y_{ji}} + \overline{x_{ji}}) = \sum_{j:(j,i)\in A} (y^p_{ji} + x^p_{ji}), \quad i = 0, 1, \ldots, n$$

then constraints (2) are satisfied.

(ii)

$$\sum_{p=1}^{P} x^p_{ij} = \frac{1}{P} \sum_{p=1}^{P} \overline{x_{ij}} \underset{(19)}{=} \frac{1}{P} \sum_{p=1}^{P} 1 \quad \forall (i,j) \in A_R$$

then constraints (3) are satisfied.

(iii)

$$\sum_{p=1}^{P} (x^p_{ij} + x^p_{ji}) = \frac{1}{P} \sum_{p=1}^{P} (\overline{x_{ij}} + \overline{x_{ji}}) \underset{(20)}{=} \frac{1}{P} \sum_{p=1}^{P} 1 = 1 \quad \forall (i,j) \in E_R$$

then (4) are satisfied.

(iv)

$$\forall\, p = 1, \ldots, P: \quad \sum_{j:(0,j)\in A} (x^p_{0j} + y^p_{0j}) = \frac{1}{P} \sum_{j:(0,j)\in A} (\overline{x_{0j}} + \overline{y_{0j}})$$
$$\underset{(21)}{\leq} \frac{P}{P} = 1$$

then restrictions (5) are satisfied.

(v)

$$\forall\, p = 1, \ldots, P: \quad \sum_{j:(j,i)\in A} f^p_{ji} - \sum_{j:(i,j)\in A} f^p_{ij} = \frac{1}{P} \left[\sum_{j:(j,i)\in A} \overline{f_{ji}} - \sum_{j:(i,j)\in A} \overline{f_{ij}} \right]$$
$$\underset{(22)}{=} \frac{1}{P} \sum_{j:(j,i)\in R} q_{ji} \overline{x_{ji}} = \frac{P}{P} \sum_{j:(j,i)\in R} q_{ji} x^p_{ji}, \quad i = 1, \ldots, n$$

then (6) are satisfied.

(vi)

$$\forall\, p = 1, \ldots, P: \quad \sum_{j:(0,j)\in A} f^p_{0j} = \frac{1}{P} \sum_{j:(0,j)\in A} \overline{f_{0j}} \underset{(23)}{=} \frac{1}{P} Q_T$$
$$= \frac{1}{P} \sum_{(i,j)\in R} q_{ij} \overline{x_{ij}} = \sum_{(i,j)\in R} q_{ij} x^p_{ij}$$

then (7) are satisfied.

(vii)

$$\forall\, p = 1, \ldots, P: \quad \sum_{i:(i,0)\in A} f^p_{i0} = \frac{1}{P} \sum_{i:(i,0)\in A} \overline{f_{i0}} \underset{(24)}{=} \frac{1}{P} \sum_{i:(i,0)\in R} q_{i0} \overline{x_{i0}}$$
$$= \sum_{i:(i,0)\in R} q_{i0} x^p_{i0}$$

then (8) are satisfied.

(viii)

$$\forall p = 1, \ldots, P: \quad f^p_{ij} = \frac{\overline{f_{ij}}}{P} \underset{(25)}{\leq} \frac{W(\overline{y_{ij}} + \overline{x_{ij}})}{P} = W(y^p_{ij} + x^p_{ij}) \quad \forall (i,j) \in A$$

then (9) are satisfied.

(ix) The nonnegativity constraints of x^p_{ij}, y^p_{ij} and f^p_{ij} are valid because the same constraints are valid on the variables $\overline{x_{ij}}$, $\overline{y_{ij}}$ and $\overline{f_{ij}}$.

Finally, the values of the two solutions are equal since:

$$Z_{LF1} = \sum_{p=1}^{P} \left[\sum_{(i,j)\in A} d_{ij} y^p_{ij} + \sum_{(i,j)\in R} c_{ij} x^p_{ij} + \lambda \sum_{(i,0)\in A} y^p_{i0} + \lambda \sum_{(i,0)\in R} x^p_{i0} \right]$$
$$= \frac{1}{P} \sum_{p=1}^{P} \left[\sum_{(i,j)\in A} d_{ij} \overline{y_{ij}} + \sum_{(i,j)\in R} c_{ij} \overline{x_{ij}} + \lambda \sum_{(i,0)\in A} \overline{y_{i0}} + \lambda \sum_{(i,0)\in R} \overline{x_{i0}} \right]$$
$$= Z_{LF2}$$

□

In the previous subsection we have shown that the linear programming relaxation of F1 can be improved by adding several sets of valid inequalities. In a similar way, the linear programming relaxation value of F2 can be improved by adding some valid inequalities. These inequalities are simply the aggregated version of the inequalities presented before.

The following inequality is the aggregated version of (13)

$$\sum_{j:(0,j)\in A} y_{0j} + \sum_{j:(0,j)\in R} x_{0j} \geq \left\lceil \frac{Q_T}{W} \right\rceil \tag{29}$$

And the constraints

$$f_{ij} \geq q_{ij} x_{ij} \quad \forall (i,j) \in R \tag{30}$$

$$f_{ij} \geq y_{ij} - P \quad \forall (i,j) \in A \backslash R \tag{31}$$

are the aggregated versions of (14) and (15).

Proposition 2 can easily be extended to the two models that include the sets of valid inequalities, F1R and F2R (model F2 reinforced with (29)–(31)).

Proposition 3. *Let* LF1R *be the linear programming relaxation of* F1R, *with optimal value* z^*_{LF1R}, *and* LF2R *the linear programming relaxation of* F2R, *with optimal value* z^*_{LF2R}. *Then* $z^*_{LF1R} = z^*_{LF2R}$.

The proof is omitted since it is similar to the proof of Proposition 2.

3. Computational results

In this section we present the results from computational experiments that were performed on a Pentium® 2.80 GHz (with 504 MB RAM) with CPLEX 11.0. In the first subsection the effect of including the valid inequalities in the proposed models is discussed. Section 3.2 compares our new approach with others from the literature. This comparison is performed on some sets of benchmark problems.

3.1. Illustrating the effect of the valid inequalities

A set of small test instances were generated to show that the inclusion of the valid inequalities in the models improve, in some cases, the linear programming relaxation bounds and/or may speed up, as a consequence, the ILP solver. These instances, named ex1–ex3, have 19–24 nodes and 29–50 links.

The results on model F1 are depicted in Table 1. Column two (F1 Opt) displays the optimum values for each example.

Table 1
Examples for the effect of the valid inequalities.

	F1 Opt	LF1 lower bounds				CPU time (s) for F1			
			(13)	(14) (15)	(16)	–	(13)	(14) (15)	(16)
ex1	3855	3738	**3850**	3738	3738	tle	0.17	tle	0.20
ex2	3985	3863	**3975**	3863	3863	tle	2.61	140.23	40.05
ex3	3865	3839	3839	**3842**	3839	3.14	3.02	2.31	2.58

Opt—represents the optimum value of F1; LF1—columns refer to the linear programming relaxation bounds; and tle—not solved in one hour (time limit exceeded).

Table 2
mval computational results.

File	$\|N\|$	$\|A \cup E\|$	$\|A_R\|$	$\|E_R\|$	P	UB	BBLP	LF1R	F1R	F2R	CPU time (s)	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	F1R	F2R
1A	24	55	20	35	4	230	**230***	203	**230***	**230***	40.45	0.17
1B	24	51	13	38	5	261	**261***	248	**261***	**261***	6.75	0.05
1C	24	53	17	36	10	315	**309**	283	292	**309**	tle	0.16
2A	24	44	16	28	4	324	**324***	315	**324***	**324***	1.09	0.03
2B	24	52	12	40	5	395	**395***	369	**395***	**395***	17.61	0.05
2C	24	49	14	35	10	526	**521**	482	488	**521**	tle	0.42
3A	24	48	15	33	4	115	**115***	111	**115***	**115***	1.63	0.03
3B	24	45	16	29	5	142	141	131	**142***	**142***	140.88	0.03
3C	24	43	18	25	9	166	**166***	147	152	163	tle	0.05
4A	41	95	26	69	5	580	**580***	552	**580***	**580***	921.59	0.05
4B	41	102	19	83	6	650	**650***	631	644	**650***	tle	0.11
4C	41	103	21	82	7	630	**630***	608	623	**630***	tle	0.06
4D	41	104	21	83	11	770	**746**	728	732	**746**	tle	1.20
5A	34	96	22	74	5	597	**597***	575	**597***	**597***	249.33	0.03
5B	34	91	35	56	6	613	**613***	578	606	**613***	tle	0.17
5C	34	98	17	81	7	697	**697***	692	**697***	**697***	244.99	0.03
5D	34	92	29	63	11	739	**719**	685	690	718	tle	2.20
6A	31	69	22	47	5	326	**326***	295	**326***	**326***	59.66	0.03
6B	31	66	22	44	6	317	**317***	296	**317***	**317***	1246.61	0.05
6C	31	68	23	45	12	371	**365**	340	340	355	tle	0.48
7A	40	86	36	50	5	364	**364***	338	363	**364***	tle	0.05
7B	40	91	25	66	6	412	**412***	406	**412***	**412***	379.61	0.05
7C	40	90	28	62	11	426	**424**	398	401	417	tle	3.45
8A	30	96	20	76	5	581	**581***	566	**581***	**581***	37.44	0.16
8B	30	91	27	64	6	531	**531***	508	**531***	**531***	2707.14	0.05
8C	30	83	28	55	11	638	**617**	590	601	**617**	tle	0.28
9A	50	132	32	100	5	458	**458***	436	454	**458***	tle	0.05
9B	50	120	44	76	6	453	**453***	428	**453***	**453***	tle	0.05
9C	50	125	42	83	7	429	**428**	396	426	**428**	tle	0.09
9D	50	131	38	93	12	520	**514**	486	487	**514**	tle	0.14
10A	50	138	32	106	5	634	**634***	613	**634***	**634***	229.39	0.05
10B	50	134	33	101	6	661	**661***	635	658	**661***	tle	0.05
10C	50	136	36	100	7	623	**623***	594	621	**623***	tle	0.06
10D	50	129	42	87	12	649	**643**	610	611	642	tle	0.30
Gap (%)		Maximum					3.29%		8.43%	4.31%	CPU	3.45
		Average					0.50%		2.13%	0.67%		0.31
No. of optimum values							23		15	23		

$|N|$—no. of nodes; $|A \cup E|$—no. of links; $|A_R|$—no. of required arcs; $|E_R|$—no. of required edges; P—no. of trips; tle—time limit exceeded; CPU time is given in seconds; UB—upper bound value from Belenguer et al. [8]; gap = [(UB−lower bound value)/UB]×100; BBLP—lower bound value from Belenguer et al. [8]; LF1R—linear programming relaxation bound; in bold represents the best lower bound value for each instance (each row); and underlined are the best lower bound obtained by CPLEX for the model.

*Represents the proven optimum of the MCARP instance.

Table 3
lpr computational results.

File	$\lvert N\rvert$	$\lvert A\cup E\rvert$	$\lvert A_R\rvert$	$\lvert E_R\rvert$	P	UB	BBLP	LF1R	F1R	F2R	CPU time (s)	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	F1R	F2R
lpr-a-01	28	94	52	0	2	13 484	**13 484***	13 484	**13 484***	**13 484***	0.84	0.03
lpr-a-02	53	169	99	5	3	28 052	**28 052***	28 037	**28 052***	**28 052***	12.83	0.03
lpr-a-03	146	469	271	33	8	76 155	76 108	75 966	76 039	**76 115**	tle	10.88
lpr-a-04	195	651	469	34	13	127 352	126 941	126 654	tle	**126 946**	tle	104.95
lpr-a-05	321	1056	748	58	20	205 499	202 735	202 259	tle	**202 736**	tle	tle
lpr-b-01	28	63	45	5	2	148 35	**14 835***	14 813	**14 835***	**14 835***	0.22	0.14
lpr-b-02	53	117	92	9	3	28 654	**28 654***	28 618	**28 654***	**28 654***	6.03	0.17
lpr-b-03	163	361	279	26	8	77 878	77 837	77 684	77 821	**77 859**	tle	0.09
lpr-b-04	248	582	493	8	15	127 454	**126932**	126 754	1267 54	**126 932**	tle	11.08
lpr-b-05	401	876	764	37	22	211 771	**209 791**	209 496	tle	209 776	tle	tle
lpr-c-01	28	52	11	39	2	18 639	**18 639***	18 445	**18 639***	**18 639***	0.89	0.25
lpr-c-02	53	101	23	77	5	36 339	**36 339***	35 899	36 255	**36 339***	tle	4.45
lpr-c-03	163	316	61	241	12	111 632	**111 117**	109 967	109 980	110 949	tle	tle
lpr-c-04	277	604	142	362	20	169 254	**168 441**	167 084	tle	168 399	tle	tle
lpr-c-05	369	841	416	387	29	259 937	**257 890**	256 287	tle	257 808	tle	tle
Gap (%)		Maximum					1.35%			1.34%	CPU	3600
		Average					0.32%			0.34%		1208.80
No. of optimum values							6		5	6		

$\lvert N\rvert$—no. of nodes; $\lvert A\cup E\rvert$—no. of links; $\lvert A_R\rvert$—no. of required arcs; $\lvert E_R\rvert$—no. of required edges; P—no. of trips; tle—time limit exceeded; CPU time is given in seconds; UB—upper bound value from Belenguer et al. [8]; gap = [(UB−lower bound value)/UB]×100; BBLP—lower bound value from Belenguer et al. [8]; LF1R—linear programming relaxation bound; in bold is represented the best lower bound value for each instance (each row); and underlined are the best lower bound obtained by CPLEX for the model.
*Represents the proven optimum of the MCARP instance.

Columns 3–6 exhibit the linear programming relaxation bounds produced by model F1 with the extra inequalities: with no additional constraints (column 3) or with the inequalities identified in the column caption introduced (columns 4–6). For each problem the best LP bound is indicated in bold. CPU times (in seconds) for the ILP solver are depicted in the last four columns, where "tle" indicates that the one hour limit was attained.

Note that the additional constraints improve the LP bounds of the original model. Adding (13) alone led to better LP bounds for ex1 and ex2 and the IP solver turns to be able to prove optimality within a few seconds (0.17 and 2.61, respectively). Concerning inequalities (14) and (15), their inclusion also conducted to improved bounds, even when compared with (13) (see ex3). The purpose of constraints (16) is just to speed up the IP solver (see columns 7 and 10).

3.2. Comparing the models with other approaches

In the present subsection we report the results from the computational tests performed in order to evaluate the quality of the new formulations in comparison with other approaches. The results were taken from two different sets of benchmark instances from the MCARP literature (see [8]).

The first set, the so-called mval files, contains the smaller instances with 24–50 nodes and 43–138 links which are all required. In these files the number of required edges is always greater than the number of required arcs.

The second set, the so-called lpr files, include 28–401 nodes and 50–1056 links and are divided into three subsets: lpr-a, lpr-b and lpr-c, with five instances in each, that differ on the percentage of required arcs and required edges included. The lpr-a and lpr-b sets of files include more required arcs than required edges, while the lpr-c files contain more required edges than arcs.

As a consequence of the preceding Section 3.1, we only report results for the enhanced models F1R and F2R and their linear programming relaxations. A time limit of one hour was also set for the tests. The lower bound values are compared with the best known lower bound (BBLP) for the tested instances and with the best known upper bound value (UB), obtained by a memetic algorithm developed by Belenguer et al. [8].

The results for mval instances are depicted in Table 2, while Table 3 refers to lpr data files. The format for both tables is the following. The first six columns (1–6), of each table, describe the instance characteristics. In column 7 upper bound values from Belenguer et al. [8] are replicated. The next four columns give lower bound values, or the optimal solution value when found, of the corresponding instance: column 8 is the lower bound provided by the method described in [8]; column 9 is the value of the linear programming relaxation of models F1R and F2R; column 10 and 11 are the best value obtained by the CPLEX after one hour CPU time of model F1R and F2R, respectively.

An (*) indicates optimal value of the MCARP. Underlined are values attained by the CPLEX that represent lower bounds for the model, whenever one hour was not enough for the IP solver to find an optimal solution. For each instance, row in the table, the best lower bound is signalized in bold.

With respect to the results for the mval instances reported in Table 2, we may notice that, the "relaxed" model F2R, column 11, produces the optimal solution values for 23 of the 34 instances, while the model F1R, column 10, only produces 15 optimums. Due to the huge number of variables, CPLEX with F1R ends up without an optimal solution and with a best lower bound that is usually worse than the value of the integer solution provided by F2R. The results of F2R are very similar to the ones of Belenguer et al. [8], referred as BBLP in (8). Their method also found 23 optimal values, differing in two cases (instances 3B and 3C) from optimums obtained by F2R. For the remaining 11 instances, the BBLP bounds are better than the F2R bounds four times.

Comparing the F2R lower bounds, column 11, with the upper bounds in 7, we may infer that the lower bounds are close to the optimal values, as gaps ((UB−LB)/UB) are no greater than 5%.

Concerning CPU times, the F2R formulation provides, in an average time of 0.31 s, never exceeding 4 s, quite reasonable bounds. For the same set of instances Belenguer et al. [8] report an average of 1.74 s, with a maximum of 24.17 s, on a 2 GHz Pentium IV.

From Table 3, where results for larger sized instances are reported, we observe again that F2R outperforms F1R. Even when an optimal solution is not produced by F2R, F2R is able to improve the value given by F1R. Comparing the reported values from columns 11 and 8, for the first class of instances, lpr-a, F2R always provides better bounds, while for the third class of instances, lpr-c, BBLP seems to be a better choice. It appears that F2R and BBLP play a complementary role, since the F2R bound is better when the number of required arcs is greater than the number of required edges, while the BBLP bounds are better when the number of required edges increases. Note also that within the lpr data, both models F2R and BBLP provide the optimum values for the smaller instances (with up to 53 nodes and 169 links).

For these instances, the lower bounds obtained by F2R deviate in no more than 2% from the upper bounds, which again indicates the quality of the bounds produced.

CPU times are quite similar to [8] and vary between few seconds for medium sized instances to the time limit for the larger instances and. As pointed out before, even in these cases, F2R was able to exhibit bounds that are better than the ones produced by BBLP, which were the best known lower bounds for the instances in this set.

4. Concluding remarks

With the aim of solving or getting good lower bounds for the MCARP, two compact models were presented. Both models are based on flows. The large number of variables and constraints of the valid formulation has motivated the development of the aggregated model. Additional constraints were added to strengthen the linear programming relaxation bounds and speed up the ILP solver.

As noted before, one of the advantages of the aggregated model lies on the fact that its linear programming relaxation value and the linear relaxation value of the disaggregated model are equal.

Computational experiments show that the bounds provided by the aggregated version are, for some benchmark instances, better than the best known lower bounds.

As far as we know, this is also the first time that a compact model for the MCARP is successfully tested on medium sized instances, with its aggregated version leading to quite promising lower bound values, in fairly small computation times.

References

[1] Amaya A, Langevin A, Trépanier M. The capacitated arc routing problem with refill points. Operations Research Letters 2007;35:45–53.

[2] Amposah SK, Salhi S. The investigation of a class of capacitated arc routing problems: the collection of garbage in developing countries. Waste Management 2004;24:711–21.

[3] Assad AA, Golden BL. Arc routing methods and applications. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. Handbooks in operations research and management science, vol. 8, network routing. Amsterdam: Elsevier; 1995. p. 375–483.

[4] Baldacci R, Maniezzo V. Exact methods based on node-routing formulations for undirected arc routing problems. Networks 2006;47:52–60.

[5] Bautista J, Fernández E, Pereira J. Solving an urban waste collection problem using ants heuristics. Computers & Operations Research 2008;35:3020–33.

[6] Belenguer JM, Benavent E. The capacitated arc routing problem: valid inequalities and facets. Computational Optimization and Applications 1998;10:165–87.

[7] Belenguer JM, Benavent E. A cutting plane algorithm for the capacitated arc routing problem. Computers & Operations Research 2003;30:705–28.

[8] Belenguer JM, Benavent E, Lacomme P, Prins C. Lower and upper bounds for the mixed capacitated arc routing problem. Computers & Operations Research 2006;33:3363–83.

[9] Corberán A, Romero A, Sanchis JM. The mixed general routing polyhedron. Mathematical Programming 2003;96:103–37.

[10] Del Pia A, Filippi C. A variable neighbourhood descent algorithm for a real waste collection problem with mobile depots. International Transactions in Operational Research 2006;13:125–41.

[11] In: Dror M, editor. Arc routing: theory, solutions and applications. Boston: Kluwer Academic Publishers; 2000.

[12] Eglese RW, Letchford AN. Polyhedral theory for arc routing problems. In: Dror M, editor. Arc routing: theory, solutions and applications. Boston: Kluwer Academic Publishers; 2000. p. 199–230.

[13] Eiselt HA, Gendreau M, Laporte G. Arc routing problems, part I: the Chinese postman problem. Operations Research 1995;43:231–42.

[14] Eiselt HA, Gendreau M, Laporte G. Arc routing problems, part II: the rural postman problem. Operations Research 1995;43:399–414.

[15] Gavish B, Graves SC. The travelling salesman problem and related problems. Working paper no. 7905, Graduate School of Management, University of Rochester; 1979.

[16] Gavish B, Graves C. Scheduling and routing in transportation and distribution systems: formulations and new relaxations. Working paper, Graduate School of Management, University of Rochester; 1982.

[17] Ghiani G, Guerreiro F, Improta G, Musmanno R. Waste collection in Southern Italy: solution of a real-life arc routing problem. International Transactions in Operational Research 2005;12:135–44.

[18] Ghiani G, Improta G, Laporte G. The capacitated arc routing problem with intermediate facilities. Networks 2001;37:134–43.

[19] Golden BL, Wong RT. Capacitated arc routing problems. Networks 1981;11:305–15.

[20] Irnich S. Solution of real-world postman problems. European Journal of Operational Research 2008;190:52–67.

[21] Lacomme P, Prins C, Ramdane-Cherif W. An integer linear model for general arc routing problems. Working paper, Université Technologie de Troyes; 2003.

[22] Longo H, Poggi de Aragão M, Uchoa E. Solving capacitated arc routing problems using a transformation to the CVRP. Computers & Operations Research 2006;33:1823–37.

[23] Magnanti TL. Combinatorial optimization and vehicle fleet planning: perspectives and prospects. Networks 1981;11:179–213.

[24] Maniezzo V. Algorithms for large directed CARP instances: urban solid waste collection operational support. Technical report UBLCS-2004-16, Department of Computer Science, University of Bologna, Italy; 2004.

[25] Moreira LM, Oliveira JF, Gomes AM, Ferreira JS. Heuristics for a dynamic rural postman problem. Computers & Operations Research 2007;34:3281–94.

[26] Mourão MC, Almeida MT. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. European Journal of Operational Research 2000;121:420–34.

[27] Mourão MC, Amado L. Heuristic method for a mixed capacitated arc routing problem: a refuse collection application. European Journal of Operational Research 2005;160:139–53.

[28] Nobert Y, Picard JC. An optimal algorithm for the mixed Chinese postman problem. Networks 1996;27:95–108.

[29] Pearn WL, Assad A, Golden BL. Transforming arc routing into node routing problems. Computers & Operations Research 1987;14:285–8.

[30] Wøhlk S. A decade of capacitated arc routing. In: Golden B, Raghavan S, Wasil E, editors. The vehicle routing problem-latest advances and new challenges, operations research/computer science interfaces. Berlin: Springer; 2008. p. 29–48.