Optim Lett
DOI 10.1007/s11590-012-0552-y

ORIGINAL PAPER

Modeling and solving the mixed capacitated general routing problem

Adamo Bosco · Demetrio Laganà · Roberto Musmanno · Francesca Vocaturo

Received: 8 September 2011 / Accepted: 1 September 2012

Abstract We tackle the mixed capacitated general routing problem (MCGRP) which generalizes many other routing problems. We propose an integer programming model for the MCGRP and extend some inequalities originally introduced for the capacitated arc routing problem (CARP). Identification procedures for these inequalities and for some relaxed constraints are also discussed. Finally, we describe a branch and cut algorithm including the identification procedures and present computational experiments over instances derived from the CARP.

Keywords Routing problem · Mixed graph · Relaxation · Separation algorithm

1 Introduction

The paper deals with the problem of routing of vehicles along the streets of residential areas to ensure a service. Each vehicle starts from the depot and comes back to it after a service trip. Street segments are modeled by edges or arcs of a network where vertices represent the intersection points of the segments. Required entities located within analyzed areas and the depot are also represented as vertices of the network.

A. Bosco · D. Laganà · R. Musmanno (✉)
Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria,
Arcavacata di Rende (CS) 87036, Italy
e-mail: musmanno@unical.it

A. Bosco
e-mail: abosco@deis.unical.it

D. Laganà
e-mail: dlagana@deis.unical.it

F. Vocaturo
Dipartimento di Economia e Statistica, Università della Calabria,
Arcavacata di Rende (CS) 87036, Italy
e-mail: vocaturo@unical.it

Published online: 27 September 2012

The literature traditionally devoted to vehicle routing problems considers two classes of problems: node routing problems (NRP_s) and arc routing problems (ARP_s). In NRP_s [16] the service activity occurs at all (or at some subsets of) the vertices, whereas in ARP_s [9,12] a single vehicle or a fleet of vehicles service all (or some subsets of) the edges and/or arcs. Although NRP_s have been studied more extensively, ARP_s have aroused a growing interest in the last two decades, prevalently among them the problem known as capacitated arc routing problem (CARP). Theoretically an ARP can be converted into an equivalent NRP [25]. However, the transformation increases the size of the instances to be solved. Consequently, most researchers prefer to study ARP_s directly. Despite the success of exact and heuristic methods for NRP_s and ARP_s, many models proposed in the literature do not give a suitable representation of real scenarios. A more general and effective class of problems is the class where the service activity occurs both at all (or at some subsets of) the vertices and at all (or at some subsets of) the edges and/or arcs. Such problems are denoted as general routing problems (GRP_s). Moreover, defining solution approaches for GRP_s is helpful because such approaches represent valid tools for solving NRP_s and ARP_s.

Orloff [22] and Lenstra and Rinnooy Kan [19] described several applications emerging in the vehicle routing field that may be modeled naturally by using GRP_s. For example, in designing routes in an urban waste collection context, the collection along a street may be modeled by means of required arcs or edges, whereas the collection occurring at specific points (e.g., hospitals, schools, and supermarkets) may be modeled by means of required vertices. In many cases, there are some restrictions for the vehicles to traverse the streets in a specified way, and the use of mixed graphs is needed. As a rule, a one-way street is represented by one arc, a two-way street in which the waste on the two sides must be collected separately is represented by two opposite arcs, and a two-way street in which the waste on the two sides can be collected in parallel and in any direction (narrow road, or very low traffic in a residential area) corresponds to one edge.

Many authors focused on the uncapacitated GRP. For example, Letchford [20,21] and Corberán and Sanchis [11] proposed a large class of valid inequalities for the GRP defined over an undirected graph. Then Corberán, Letchford and Sanchis [5] described a cutting plane algorithm based on facet-inducing inequalities. The first remarkable contribution focused on the GRP defined over a mixed graph was proposed by Corberán, Romero and Sanchis [10]. They presented a cutting-plane algorithm, subsequently improved by Corberán, Mejía and Sanchis [6]. Corberán, Plana and Sanchis [7,8] referred to windy graphs. In particular, they presented a windy general routing polyhedron description and designed a branch and cut algorithm able to solve optimally a quite large number of instances.

This paper refers to the capacitated case. Specifically, we deal with the mixed capacitated general routing problem (MCGRP), i.e., the problem of finding a set of vehicle routes over a mixed graph such that each route starts and ends at the depot, with each required vertex/arc/edge serviced by exactly one vehicle, while the total demand serviced by each vehicle does not exceed its capacity, and the total traveling cost is minimized. Many routing problems are special cases of the MCGRP, like the CARP and the uncapacitated GRP defined over directed, undirected and mixed graphs. Since the MCGRP includes a large element of NP-hard problems, it is also an

NP-hard problem. To our knowledge, the MCGRP has been exclusively tackled by means of heuristic and metaheuristic approaches. Pandit and Muralidharan [24] proposed a heuristic procedure named ROUTE1 that starts with a condensed sub-graph obtained from the original network by considering only the required arcs, edges and vertices. Since the sub-graph is generally disconnected, the connection is reached by adding to it the shortest paths linking two vertices of disjoint connected components. The sub-graph is then converted into an Eulerian graph which admits a giant tour. A feasible solution is obtained by cutting the giant tour into smaller tours satisfying the capacity constraints. Gutiérrez, Soler and Hervás [18] introduced an alternative constructive procedure, based on the partition-first-route-next paradigm, improving previous results. Finally, Prins and Bouchenoua [27] described a memetic algorithm for the MCGRP. Three suitable procedures, named nearest neighbor heuristic, merge heuristic and tour splitting heuristic, were defined to initialize their metaheuristic algorithm.

The main contribution of this work consists in studying the MCGRP through an integer linear programming model. The remainder of the paper is organized as follows: in Sect. 2 we introduce a mathematical formulation for the MCGRP; in Sect. 3 we discuss relaxations of the formulation and valid inequalities for the problem and report exact and/or heuristic procedures for their identification; a branch and cut (B&C) algorithm is illustrated in Sect. 4; Sect. 5 shows computational results and provides final remarks.

2 Mathematical model

Let $G = (V, A, E)$ be a mixed graph defined by a set of vertices $V = \{1, \dots, n\}$, a set of arcs $A = \{(i, j) \subseteq V \times V\}$ and a set of edges $E = \{(i, j) \subseteq V \times V : i < j\}$. Vertex $1 \in V$ represents the depot at which m identical vehicles of capacity Q are based. Each element of $A \cup E$ will be referred in the following as *link*, while the set of vertices different from the depot will be denoted by $C = V \setminus \{1\}$. Some subsets of arcs and edges, denoted respectively by $A_R \subseteq A$ and $E_R \subseteq E$, are *required*, i.e., they must be *serviced* by one vehicle, but any link of $A \cup E$ can be *deadheaded* any number of times. Similarly, a subset $V_R \subseteq C$ of *required* vertices needs to be serviced by one vehicle. Required links and vertices cannot be split. Each link $(i, j) \in A \cup E$ has a non-negative cost c_{ij}. In addition, each required link $(i, j) \in A_R \cup E_R$ has a non-negative demand d_{ij}, and each required vertex $i \in V_R$ has a non-negative demand q_i. In order to ensure feasibility, we assume that the demand of each required link and vertex does not exceed Q. Note that there is a graph G^R induced on G by all the required links and vertices. Generally, this graph is non-connected. The vertex sets corresponding to connected components of G^R are called R-sets. The subgraphs of G induced by the R-sets define the so-called R-connected components of G. Observe that every isolated required vertex represents an R-connected component of G.

In the following, further notation used throughout the paper is introduced. Given a subset $S \subset V$ of vertices, then $\overline{S}$ denotes its complementary set ($\bar{S}=V \setminus S$). Let $\delta^+(S) = \{(i, j) \in A : i \in S \wedge j \in \overline{S}\}$ be the set of arcs leaving S, $\delta^-(S) = \{(i, j) \in A : i \in \overline{S} \wedge j \in S\}$ the set of arcs entering S, $\delta_R^+(S) = \{(i, j) \in A_R : i \in S \wedge j \in \overline{S}\}$ the set of required arcs leaving S, $\delta_R^-(S) = \{(i, j) \in A_R : i \in \overline{S} \wedge j \in S\}$ the set of required arcs entering S, $\delta(S) = \{(i, j) \in E : i \in S \wedge j \in \overline{S}, \text{ or } i \in \overline{S} \wedge j \in S\}$ the set of edges incident to S, and $\delta_R(S) = \{(i, j) \in E_R : i \in S \wedge j \in \overline{S}, \text{ or } i \in \overline{S} \wedge j \in S\}$ the set of required edges incident to S. Whenever $S = \{v\}$ the previous notation remains valid as long as S is replaced by v, and $\overline{S}$ by $\overline{v}$, or $V\backslash\{v\}$. Moreover, let $S_R = S \cap V_R$ be the set of required vertices belonging to S, $A_R(S) = \{(i, j) \in A_R : i \in S \wedge j \in S\}$ the set of required arcs with both endpoints in S, and $E_R(S) = \{(i, j) \in E_R : i \in S \wedge j \in S\}$ the set of required edges with both endpoints in S.

We propose an integer linear programming formulation based on three-index link variables and two-index vertex variables. For a required link (i, j) and a vehicle k, let x_{ij}^k be a binary variable equal to 1 if and only if (i, j) is serviced by vehicle k which travels from vertex i to vertex j. For a link (i, j) and a vehicle k, let y_{ij}^k be a non-negative variable representing the number of deadheading from vertex i to vertex j by k. Finally, for a required vertex i and a vehicle k, let z_i^k be a binary variable equal to 1 if and only if i is serviced by k. Using these variables, the MCGRP can be formulated as follows.

$$\begin{aligned}\text{Min } \lambda = &\sum_{k\in K}\sum_{(i,j)\in E_R} c_{ij}(x_{ij}^k + x_{ji}^k) + \sum_{k\in K}\sum_{(i,j)\in A_R} c_{ij}x_{ij}^k \\ &+ \sum_{k\in K}\sum_{(i,j)\in E} c_{ij}(y_{ij}^k + y_{ji}^k) + \sum_{k\in K}\sum_{(i,j)\in A} c_{ij}y_{ij}^k \end{aligned} \tag{1a}$$

$$\sum_{k\in K}(x_{ij}^k + x_{ji}^k) = 1, \quad \forall (i, j) \in E_R \tag{1b}$$

$$\sum_{k\in K} x_{ij}^k = 1, \quad \forall (i, j) \in A_R \tag{1c}$$

$$\sum_{k\in K} z_i^k = 1, \quad \forall i \in V_R \tag{1d}$$

$$\sum_{(i,j)\in E_R} d_{ij}(x_{ij}^k + x_{ji}^k) + \sum_{(i,j)\in A_R} d_{ij}x_{ij}^k + \sum_{i\in V_R} q_i z_i^k \le Q, \quad \forall k \in K \tag{1e}$$

$$\begin{aligned}&\sum_{j:(i,j)\in\delta_R^+(i)} x_{ij}^k + \sum_{j:(i,j)\in\delta^+(i)} y_{ij}^k - \sum_{j:(j,i)\in\delta_R^-(i)} x_{ji}^k - \sum_{j:(j,i)\in\delta^-(i)} y_{ji}^k = \\ &\sum_{j:(i,j)\in\delta_R(i)} x_{ji}^k + \sum_{j:(i,j)\in\delta(i)} y_{ji}^k - \sum_{j:(i,j)\in\delta_R(i)} x_{ij}^k - \sum_{j:(i,j)\in\delta(i)} y_{ij}^k, \quad \forall k \in K,\ i \in V\end{aligned} \tag{1f}$$

$$\begin{aligned}&\sum_{(i,j)\in\delta_R^+(S)} x_{ij}^k + \sum_{(j,i)\in\delta_R^-(S)} x_{ji}^k + \sum_{(i,j)\in\delta_R(S)} (x_{ij}^k + x_{ji}^k) + \sum_{(i,j)\in\delta^+(S)} y_{ij}^k \\ &+ \sum_{(j,i)\in\delta^-(S)} y_{ji}^k + \sum_{(i,j)\in\delta(S)} (y_{ij}^k + y_{ji}^k) \\ &\ge \begin{cases} 2(x_{uv}^k + x_{vu}^k), & \forall (u, v) \in E_R(S), \\ 2x_{uv}^k, & \forall (u, v) \in A_R(S), \\ 2z_h^k, & \forall h \in S_R, \\ & \quad k \in K, \quad S \subseteq C \end{cases}\end{aligned} \tag{1g}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall k \in K, \ (i, j) \in A_R \cup E_R \tag{1h}$$

$$x_{ji}^k \in \{0, 1\}, \quad \forall k \in K, \ (i, j) \in E_R \tag{1i}$$

$$y_{ij}^k \in \mathcal{Z}_+, \quad \forall k \in K, \ (i, j) \in A \cup E \tag{1j}$$

$$y_{ji}^k \in \mathcal{Z}_+, \quad \forall k \in K, \ (i, j) \in E \tag{1k}$$

$$z_i^k \in \{0, 1\}, \quad \forall k \in K, \ i \in V_R. \tag{1l}$$

The objective function (1a) minimizes the total routing cost. Constraints (1b)–(1d) ensure that each request is serviced exactly once by exactly one vehicle (*assignment constraints*). Constraints (1e) model the demand limitations imposed by the capacity Q of each vehicle (*knapsack constraints*). They provide the connection between the scheduling and routing structure of the MCGRP polyhedron. Inequalities (1f) represent *flow constraints*. They model the symmetry conditions at each vertex. Note that, together with the integrality conditions, constraints (1f) also imply parity conditions at each vertex. Constraints (1g) are *connectivity constraints*. They impose that for each subset of vertices (excluding the depot) containing a required link or vertex serviced by a vehicle, at least two links incident to the subset must be used to visit it (deadheaded or serviced); they also eliminate subtours disjointed from the depot. Finally, constraints (1h)–(1l) define variable domains. Model (1) is quite complex and can be optimally solved only for instances with a small number of vehicles. Anyway, this is not a limitation for many transport operators, whose fleet has a limited size. Note that our formulation can be easily adapted in order to consider windy graphs.

3 Relaxations and valid inequalities

Owing to the exponential number of connectivity constraints that involve a very large number of subsets, only a limited number of such constraints is considered into an initial relaxation of the model. We use the separation routines described in Section 3.1 to find violations. Moreover, we extend some well-known inequalities already introduced in [1] and [2] for a surrogate CARP formulation to the MCGRP polyhedron. More details are given in Sect. 3.2.

3.1 Connectivity constraints: separation algorithms

Firstly, connectivity constraint violations are checked through a heuristic algorithm, although the separation problem is solvable in polynomial time. We use a modification of the heuristic procedure proposed in [14]. Given a solution of the relaxed MCGRP model, three vectors of variables are defined for a vehicle index k. Specifically, x^k is a $(|A_R| + 2|E_R|)$-dimensional vector only including variables x_{ij}^k associated with k, y^k is a $(|A| + 2|E|)$-dimensional vector only including variables y_{ij}^k associated with k, and z^k is a $(|V_R|)$-dimensional vector defined by variables z_i^k associated with k. Let $(\bar{x}^k, \bar{y}^k, \bar{z}^k)$ be the optimal solution of the linear programming relaxation referring to the k-th route. For each k, let $C_1^k, \ldots, C_\rho^k$ be the connected components induced on G by the links $(i, j) \in A \cup E$ such that $\bar{x}_{ij}^k > 0$ and $\bar{y}_{ij}^k > 0$, and the vertices $h \in V$ such that $\bar{z}_h^k > 0$. Moreover, let $V_1^k, \ldots, V_\rho^k$ be the vertex sets corresponding to these connected components of G, and G^k the auxiliary graph where each vertex is associated to $C_i^k = G(V_i^k),\ i = 1, \ldots, \rho$. Each pair of vertices of G^k is linked by an edge (s, r) whose cost is the sum of variables $\bar{x}_{ij}^k$ and $\bar{y}_{ij}^k$ corresponding to edges $(i, j) \in A \cup E$ such that $i \in V_s^k$ and $j \in V_r^k$, for each $s = 1, \ldots, \rho$ and $r = 1, \ldots, \rho$ with $s \neq r$. We use Prim's algorithm [26] to construct a maximum spanning tree on G^k. At any stage of its construction, let S^k be the set of connected components corresponding to the vertices of the partial tree. If S^k yields a violated connectivity constraint, it is generated. Once the spanning tree is complete, another check for violations is made by removing in turn each edge of the tree.

An exact algorithm comes into play whenever the heuristic algorithm fails. Our exact separation algorithm follows the outline provided in [4] for the CARP. Specifically, for each vehicle index k, let $G^k(w)$ be an undirected graph including the depot and induced by the edges $(i, j) \in E$ with a capacity defined by $w_{ij}^k = \bar{x}_{ij}^k + \bar{x}_{ji}^k + \bar{y}_{ij}^k + \bar{y}_{ji}^k > 0$ and the edges corresponding to the arcs $(i, j) \in A$ with a capacity defined by $w_{ij}^k = \bar{x}_{ij}^k + \bar{y}_{ij}^k > 0$. Constraints (1g) can be separated in polynomial time by solving a min-cut separating the depot from each vertex of $G^k(w)$.

3.2 Surrogate inequalities: definition and identification

Belenguer and Benavent [1,2] observed a considerable improvement in the lower bound for the CARP by aggregating three-index variables with respect to the indices of the vehicles. In accordance with their works, we defined valid inequalities for our problem. Let $G^w = (V^w, E^w)$ be a windy graph obtained from G by replacing each arc (i, j) with an edge. This transformation helps to make our problem closer to the original one by Belenguer and Benavent. The cost on a new edge of G^w is the same as the cost of the correspondent arc (i, j) in the original graph if the edge is traversed from i to j, it is ∞ otherwise. Note that $V^w = V$ and $C^w = C$. The notation introduced in Sect. 2 can be transposed to G^w. Therefore, E_R^w denotes the set of required edges, V_R^w denotes the set of required vertices, $\delta^w(S)$ and $\delta_R^w(S)$ denote, respectively, the (cut)sets of edges and required edges with one endpoint in S and the other one not in S, S_R^w denotes the set of required vertices in S, and $E_R^w(S)$ denotes the set of required edges with both endpoints in S. Let θ_{ij} be an integer variable representing the total number of times that $(i, j) \in E^w$ is deadheaded by the vehicles. We say that a vehicle *crosses* a cutset $\delta^w(S)$ whenever it traverses an edge $(i, j) \in \delta^w(S)$. The so-called *capacity constraints* are valid inequalities that may be expressed in terms of the aggregated variables θ_{ij}:

$$\sum_{(i,j)\in\delta^w(S)} \theta_{ij} \geq 2\eta(S) - |\delta_R^w(S)| \quad \forall S \subseteq C^w, \tag{2}$$

where $\eta(S) = \lceil (\sum_{(i,j)\in E_R^w(S)\cup\delta_R^w(S)} d_{ij} + \sum_{i\in S_R^w} q_i)/Q \rceil = \lceil D(S)/Q \rceil$. Note that at least $\eta(S)$ vehicles are needed to service all the required edges in the cutset $\delta_R^w(S)$

and inside S, as well as the required vertices in S. In fact, a vehicle which services some required edges in $E_R^w(S) \cup \delta_R^w(S)$ and/or some required vertices in S_R^w crosses $\delta^w(S)$ at least twice, so the number of deadheaded crossings of the cutset $\delta^w(S)$ is at least $2\eta(S) - |\delta_R^w(S)|$. The solution graph associated with the feasible solutions of the MCGRP must be an even graph, i.e., all its vertices must have an even degree. It can be easily shown that a cutset must contain an even number of edges. For each cutset containing an odd number of required edges, at least one edge in the cut must be deadheaded. This fact is expressed by the so-called *odd edge cutset constraints*:

$$\sum_{(i,j)\in\delta^w(S)} \theta_{ij} \geq 1 \quad \forall S \subseteq C^w, \quad \text{with } |\delta_R^w(S)| \text{ odd.} \tag{3}$$

Constraints (2) and (3) can be rewritten in the following unified way:

$$\sum_{(i,j)\in\delta^w(S)} \theta_{ij} \geq \alpha(S) \quad \forall S \subseteq C^w, \tag{4}$$

where $\alpha(S)$ is equal to $\max\{2\eta(S) - |\delta_R^w(S)|, 1\}$ if $|\delta_R^w(S)|$ is odd, and to $\max\{2\eta(S) - |\delta_R^w(S)|, 0\}$ if $|\delta_R^w(S)|$ is even.

In order to express (4) in terms of the original variables of the MCGRP model, we observe that θ_{ij} is equal to $\sum_{k\in K}(y_{ij}^k + y_{ji}^k)$ if (i, j) is an edge in G (i.e., $c_{ji} = c_{ij}$ in G^w), and to $\sum_{k\in K} y_{ij}^k$ if (i, j) is an arc in G (i.e., $c_{ji} = \infty$ in G^w).

The sets in the formulas, e.g. $\delta_R^w(S)$, can be easily transformed with respect to the original graph G. We denote by $\bar{\theta}_{ij}$ the current optimal value for the aggregated variable θ_{ij}. It is not known whether the separation problem of the capacity constraints (2) is $NP-$hard or not. However, the so-called *fractional capacity constraints* can be separated in polynomial time. They are as follows:

$$\sum_{(i,j)\in\delta^w(S)} \theta_{ij} \geq 2\left(\frac{D(S)}{Q}\right) - |\delta_R^w(S)| \quad \forall S \subseteq C^w. \tag{5}$$

Since $\eta(S) \geq D(S)/Q$, inequalities (2) dominate inequalities (5). The fractional capacity constraints can be effectively identified by using a procedure similar to the one described in [2]. It consists of solving a maximum flow problem on a graph $\bar{G}^w$ obtained from G^w by adding an artificial vertex σ and edges connecting σ to the other vertices in G^w. The capacity of each edge (i, j) in $\bar{G}^w$ is denoted by b_{ij}. Particularly, b_{ij} is equal to $\bar{\theta}_{ij}$ if $(i, j) \in E^w \backslash E_R^w$, to $\left(\bar{\theta}_{ij} + 1 - \frac{d_{ij}}{Q}\right)$ if $(i, j) \in E_R^w$, and to $\left(\frac{2}{Q}q_i + \frac{1}{Q}\sum_{(i,h)\in\delta_R^w(i)} d_{ih}\right)$ if $i \in V^w$ and $j = \sigma$. Observe that in the last expression, b_{ij} drops to $\left(\frac{1}{Q}\sum_{(i,h)\in\delta_R^w(i)} d_{ih}\right)$ whenever $i \notin V_R^w$. Let υ be the minimum capacity of the cut defined by $S \cup \{\sigma\}$ and obtained by solving a maximum flow problem on $\bar{G}^w$ between vertices 1 and σ. Note that S represents the set of original vertices defining this optimal cut. Let us define P as $\frac{2}{Q}(\sum_{(i,j)\in E_R^w} d_{ij} + \sum_{i\in V_R^w} q_i)$. The slack of constraint (5) for S is obtained by subtracting P to υ. It can be easily shown that the operation $\upsilon - P$ provides the following result:

$$\sum_{(i,j)\in\delta_R^w(S)} b_{ij} + \sum_{(i,j)\in\delta^w(S)\setminus\delta_R^w(S)} b_{ij} + \sum_{i\in V^w\setminus S} b_{i\sigma} - \frac{2}{Q}\sum_{(i,j)\in E_R^w} d_{ij} - \frac{2}{Q}\sum_{i\in V_R^w} q_i$$
$$= \sum_{(i,j)\in\delta^w(S)} \bar{\theta}_{ij} + |\delta_R^w(S)| - 2\frac{D(S)}{Q}.$$

Therefore, if $\upsilon - P < 0$ then constraints (5) and (2) are violated for S. If $\upsilon - P \geq 0$ then no constraint of type (5) is violated, but (2) could be violated for S.

Odd edge cutset constraints can be separated exactly in polynomial time by adapting the procedure of Padberg and Rao [23], or heuristically through the procedure described in [5]. Nevertheless, in our algorithm odd edge cutset constraints are generated only for some subsets of vertices and added to the initial linear relaxation to strengthen lower bounds. In effect, during the experimental phase, their contribution appeared negligible.

4 The overall algorithm

Five procedures were used in order to obtain an initial MCGRP feasible solution. Three procedures are based on the partition-first-route-next methodology. In particular, the first procedure follows the outline provided in [18]. The fourth procedure is based on the path scanning methodology [15]. Specifically, the method described in [28] is adapted to the MCGRP. The latter procedure implements the feasibility pump scheme [13]. For all tested instances, described in Sect. 5, the feasibility pump-based procedure outperforms the other heuristics. For this reason, our B&C always starts from the solution provided by this procedure whose value represents an upper bound on the optimal value.

The efficiency of the B&C algorithm also depends on the strategy used to strengthen the relaxation at the root-node and obtain a good lower bound. The initial relaxation includes: the objective function (1a), the constraints (1b), (1c), (1d), (1e), (1f), and the connectivity constraints (1g) associated with the R-sets. It is reinforced through the procedure briefly described in the following.

Root-node procedure. It includes odd edge cutset inequalities (3) generated for the sets $S = \{i\}$, where i is an odd vertex, i.e., a vertex incident with an odd number of required edges. It also includes other constraints (1g), other inequalities (3) and inequalities (2) generated by the following scheme:

```
set W = {1}
while W ≠ V do
   set S = V \ W, generate constraint (1g), and compute α(S)
   if α(S) > 0 then
      generate inequality (2) or (3)
   end if
   Add to W those vertices adjacent to at least a vertex of W
end while.
```

Another key aspect in the B&C algorithm is the so-called *cut pool management*. Specifically, an iteration of the B&C algorithm involves the selection of a subproblem from the list of active subproblems and the addition of violated constraints and valid inequalities to this subproblem. The set containing violated constraints and valid inequalities is called cut pool. It is cleaned every 50 iterations by eliminating inequalities with slack variables more than ϵ or dual variables less than ϵ, where $\epsilon = 10^{-6}$ is a tolerance.

An outline of the B&C algorithm is provided in the following (the cut pool cleaning is not included in order to simplify the scheme).

- Step 1. Obtain an upper bound $\bar{\lambda}$ on the optimal solution value λ^*.
- Step 2. Define a relaxed MCGRP formulation by considering constraints (1g) only for the R-sets and eliminating integrality constraints.
- Step 3. Reinforce the previous linear program by calling the root-node procedure. Insert the resulting subproblem in a list L.
- Step 4. If L is empty, STOP. Otherwise extract from L a subproblem.
- Step 5. Solve the subproblem. Let λ be the solution value. If $\lambda \geq \bar{\lambda}$, go back to step 4.
- Step 6. Identify constraints (1g). If the heuristic algorithm for the identification of the connectivity constraints fails, i.e., it finds no violation, apply the exact separation algorithm.
- Step 7. Identify heuristically inequalities of type (2). If no inequalities of type (1g) and (2) has been identified in steps 6 and 7, set $\bar{\lambda} = \lambda$ and go back to step 4 if the current solution is feasible, otherwise go to step 8. If some violated inequalities (1g) and (2) have been identified in steps 6 and 7, add these inequalities to the cut pool and go back to step 5.
- Step 8. Generate two subproblems by branching on a fractional variable. Select the branching variable by considering the following order: $x_{ij}^k, z_i^k, y_{ij}^k$. Insert the subproblems in L and go back to step 4.

5 Results and discussion

Computational experiments were carried out on a PC equipped with 2 Intel Xeon Quad Core CPUs @3.0 GHz, with 6 Gbyte RAM. The B&C algorithm was coded in *java*, by using ILOG CPLEX library, release 12.2 and activating all the standard CPLEX cuts. Note that the feasibility pump scheme to obtain an initial solution is already implemented in this version of CPLEX.

We tested our B&C algorithm on different datasets derived from existing datasets. The first class is derived from the *gdb* instances introduced for the undirected CARP [15]. With the aim of generating mixed and general problems from the *gdb* tests, we modified them in the following way. Firstly, we replaced $\lceil 0.75|E|\rceil$ edges with pairs of opposite arcs and moved the demand of each required edge to one (randomly chosen) of the two arcs as soon as that edge is replaced. Then, we designed six different datasets from each modified problem by shifting the demands of $\lceil \beta\pi \rceil$ randomly selected required links to $\lceil \beta\pi \rceil$ randomly selected adjacent vertices, where π is the number of required links in our mixed graphs, and assigned to β the values in the set $\{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$. Observe that the shifting was performed by checking that the demand of each required link and vertex does not exceed the capacity of the vehicle. The resulting set is composed of 114 instances called *mggdb*, where we extend the original acronym *gdb* with *m*, which means *mixed*, and *g*, which means *general*. The second class of datasets is derived from the *mval* dataset designed for the mixed CARP [3]. Similarly to *gdb* instances, we transformed the *mval* problems to get general problems, by shifting the demand from the required links to the vertices. The resulting set is composed of 150 instances, each of them referred to as *mgval* (*g* means, as usual, *general*). Note that we limit the computational investigation to those problems with a number of vehicles $m \leq 7$ because of the complexity of the three-index formulation. Anyway, instances with up to seven vehicles represent a benchmark with an intermediate difficulty level in the context of the capacitated routing problems. A time limit of 6 h was imposed to the computations, so that valid lower and upper bounds for the MCGRP were obtained whenever the algorithm stopped without satisfying the termination criterion (the optimality gap provided by CPLEX is equal to 0). Numerical results are reported in Tables 1–3. The column headings are defined as follows: FILE denotes the instance name; m denotes the number of vehicles; $|V|$ denotes the number of vertices; $|A|$ denotes the number of arcs; $|E|$ denotes the number of edges; $|V_R|$ denotes the number of required vertices; $|A_R|$ denotes the number of required arcs; $|E_R|$ denotes the number of required edges; LB denotes the lower bound at the root of the search tree; CON denotes the number of added connectivity constraints; SUR denotes the number of added surrogate capacity inequalities; λ denotes the best solution value reached within the time limit (an optimal value is marked by an asterisk); GAP denotes the percentage gap provided by CPLEX (if the model is solved to optimality, it is equal to 0); NODES denotes the number of nodes explored in the search tree; TIME denotes the computational time in seconds (the time limit is represented by –).

With respect to the *mggdb* datasets, the number of instances solved to optimality is equal to 12 with $\beta = 0.25$, 13 with $\beta \in \{0.30, 0.45\}$, 16 with $\beta = 0.35$, and 15 with $\beta \in \{0.40, 0.50\}$; the average percentage gaps are equal to 1.39 with $\beta = 0.25$, 1.33 with $\beta = 0.30$, 0.59 with $\beta = 0.35$, 0.98 with $\beta = 0.40$, 1.87 with $\beta = 0.45$, and 1.37 with $\beta = 0.50$. With respect to the *mgval* datasets, the number of instances solved to optimality is equal to 12 with $\beta \in \{0.25, 0.35\}$, 9 with $\beta = 0.30$, 14 with $\beta = 0.40$, 10 with $\beta = 0.45$, and 13 with $\beta = 0.50$; the average percentage gaps are equal to 1.75 with $\beta = 0.25$, 2.04 with $\beta = 0.30$, 2.11 with $\beta = 0.35$, 2.01 with $\beta = 0.40$, 2.56 with $\beta = 0.45$, and 2.29 with $\beta = 0.50$. In general, the average gaps for the *mgval* instances are slightly higher than the average gaps for the *mggdb* instances. These results can depend on the different structure of the graphs. The tables show that the possibility of finding the optimal solution decreases with the increase of m. In fact, instances with a large number of vehicles are difficult to solve, especially owing to the increase in the number of feasible and equivalent solutions relating to the assignment of the vehicles to the required links and vertices. Moreover, we note that the increase in the number of required vertices with respect to the number of required links, obtained by changing β, does not affect the solvability of the instances. This may be explained by observing that the number of isolated required vertices can increase with respect to the total number of required vertices, by leading to the creation of more R-connected components and connectivity cuts. Finally, we observe that the contribution of the

Table 1 Computational results for the datasets with $\beta \in \{0.25, 0.30\}$

File	m	$\|V\|$	$\|A\|$	$\|E\|$	$\beta = 0.25$										$\beta = 0.30$									
					$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME
mggdb19	3	8	18	2	3	6	1	47	141	2	53*	0	49	1.14	3	6	1	50	0	0	51*	0	5	0.36
mggdb10	4	12	38	6	4	14	4	265	141	0	265*	0	40	2.79	5	13	4	228	432	66	242*	0	557	12.06
mggdb15	4	7	32	5	5	12	3	55	2	0	55*	0	1	0.47	5	11	3	44	22	0	44*	0	1	0.48
mggdb20	4	11	34	5	5	12	3	116	459	0	116*	0	261	7.8	4	11	3	91	1,028	204	94*	0	1,856	27.27
mggdb4	4	11	30	4	4	11	3	272	620	249	289*	0	1,909	22.68	6	10	2	231	883	25	260*	0	3,796	39.7
mggdb1	5	12	34	5	6	12	3	251	1,180	2,470	280*	0	136,884	2,079.41	7	11	3	238.25	1,358	60,735	273*	0	273,340	4,279.45
mggdb14	5	7	32	5	5	12	3	107	7	0	107*	0	12	1.43	3	11	3	98	326	0	101*	0	21,609	184.67
mggdb16	5	8	42	7	5	15	5	98	226	0	98*	0	97	4.99	6	14	4	105	144	0	105*	0	21	2.77
mggdb17	5	8	42	7	5	15	5	71	20	0	71*	0	1	1.02	4	14	4	65	7	0	65*	0	1	0.64
mggdb3	5	12	34	5	7	12	3	253.5	1,582	4,993	278*	0	60,256	923.88	5	11	3	253	1,260	460	270*	0	287,271	4,447.9
mggdb6	5	12	34	5	6	12	3	284	1,224	35	292*	0	790	14.89	8	11	3	235	1,073	348	276*	0	30,075	443.39
mggdb7	5	12	34	5	5	12	3	275	763	48	290*	0	2,284	40.35	6	11	3	228.5	1,971	2,253	273*	0	174,148	2,870.34
mggdb11	5	22	68	11	8	25	8	345	17,001	2,575	356	3.09	163,442	–	13	23	7	381	15,805	24,089	387	1.55	139,578	–
mggdb18	5	9	54	9	6	20	6	139	5,210	0	144	3.47	659,086	–	6	18	6	144	31	0	144*	0	10	1.59
mggdb21	6	11	50	8	7	18	6	145	1,971	0	146	0.68	494,732	–	6	17	5	120	2,060	104	121	0.83	541,502	–
mggdb13	6	10	42	7	6	15	5	374	3,065	2,138	388	3.38	563,260	–	6	14	4	447	4,964	6,003	486	8.02	155,795	–
mggdb5	6	13	40	6	5	15	4	364	3,322	76,851	394	6.41	502,821	–	7	14	4	369	3,259	10,458	388	2.82	708,515	–
mggdb2	6	12	40	6	6	15	4	317	3,019	58,526	349	5.58	608,366	–	6	14	4	270	3,484	17,792	301	5.74	609,011	–
mggdb12	7	13	36	5	6	13	3	400	3,561	20,841	459	3.74	490,959	–	6	12	3	395	5,098	53,866	467	6.36	467,713	–

Table 1 continued

File	m	$\|V\|$	$\|A\|$	$\|E\|$	$\beta = 0.25$										$\beta = 0.30$									
					$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME
mgval1A	2	24	35	20	13	26	15	177	1,284	0	177*	0	1	0.3	15	24	14	168	878	0	170*	0	21	4.38
mgval2A	2	24	28	16	7	21	12	251	1,021	16	259*	0	127	6.1	12	19	11	228.44	445	41	233*	0	29	1.58
mgval3A	2	24	33	15	9	24	11	88	4,111	14	89*	0	26	3.21	13	23	10	103.5	2,899	20	105*	0	86	6.34
mgval10A	3	50	106	32	26	79	24	492	23,679	52	492*	0	1	5.08	31	74	22	481	240,376	2,014	484	0.62	22,951	–
mgval1B	3	24	38	13	10	28	9	217	481	31	217*	0	1	0.13	12	26	9	182	4,541	357	194*	0	333	24.46
mgval2B	3	24	40	12	9	30	9	318	6,495	11,873	336*	0	21,797	941.47	13	28	8	322.83	14,846	32,601	347*	0	68,708	3,309.99
mgval3B	3	24	29	16	8	21	12	112.5	2,348	96,086	125*	0	172,610	7,217.32	10	20	11	108	654	544	115*	0	764	31.27
mgval4A	3	41	69	26	19	51	19	504	193,703	47,059	514	1.55	55,887	–	21	48	18	466	148,091	28,929	477	1.75	37,455	–
mgval5A	3	34	74	22	21	55	16	483	95,220	5,091	485*	0	6,962	2,418.17	20	51	15	444	29,520	318	445*	0	281	96.32
mgval6A	3	31	47	22	16	35	16	274	9,337	0	274*	0	19	11.81	19	32	15	240	35,846	76,717	252	1.84	139,340	–
mgval7A	3	40	50	36	20	37	27	294.5	24,156	0	297*	0	357	121.29	17	35	25	324	8,246	0	324*	0	1	1.72
mgval8A	3	30	76	20	16	57	15	507	83,018	188	510	0.39	100,543	–	21	53	14	427.5	104,802	40,201	431	0.81	71,615	–
mgval9A	3	50	100	32	23	75	24	367	253,845	0	371	1.08	9,174	–	26	70	22	356	187,748	0	357	0.28	13,960	–
mgval10B	4	50	101	33	24	75	24	528	48,721	10	528*	0	1	1.64	30	70	23	435	207,306	6,034	441	1.36	9,263	–
mgval4B	4	41	83	19	20	62	14	506.25	188,117	17,583	537	4.74	18,239	–	27	58	13	491	130,151	33,829	533	5.19	35,326	–
mgval5B	4	34	56	35	18	42	26	472	86,202	6,155	493	4.26	19,222	–	20	39	24	465	92,965	25,056	490	4.58	35,208	–
mgval6B	4	31	44	22	14	33	16	253.5	28,574	36,663	263	1.93	97,822	–	19	30	15	251	31,089	2,811	262*	0	42,748	6,331.62
mgval7B	4	40	66	25	18	49	18	351	68,769	4,033	355*	0	5,821	1,859.86	19	46	17	336	49,345	3,469	344	1.91	42,695	–
mgval8B	4	30	64	27	16	48	20	405	206,425	18,157	423	4.26	18,244	–	21	44	18	390.5	151,566	18,504	400	2.35	19,830	–
mgval9B	4	50	76	44	22	57	33	354	182,229	10,164	358	1.12	10,517	–	27	53	30	344	157,078	12,246	348	1.15	14,870	–
mgval10C	5	50	100	36	27	75	27	480	297,200	5,125	483	0.62	2,797	–	30	70	25	464	302,338	4,227	478	2.93	2,538	–
mgval4C	5	41	82	21	24	61	15	504	148,773	11,688	525	4	10,352	–	27	57	14	468.5	119,530	15,546	498	5.92	14,205	–
mgval5C	5	34	81	17	21	60	12	561	101,325	4,258	584	3.94	18,528	–	20	56	11	513	98,376	12,115	551	6.8	28,871	–
mgval9C	5	50	83	42	26	62	31	347	260,197	3,694	365	4.93	2,485	–	25	58	29	330	138,262	7,517	335	1.49	6,351	–
mgval3C	7	24	25	18	10	18	13	129.5	12,715	114,954	153	11.01	146,575	–	12	17	12	126.5	14,869	92,280	153	12.08	118,018	–

Table 2 Computational results for the datasets with $\beta \in \{0.35, 0.40\}$

File	m	$\|V\|$	$\|A\|$	$\|E\|$	$\beta = 0.35$										$\beta = 0.40$									
					$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME
mggdb19	3	8	18	2	3	5	1	51	5	2	51*	0	1	0.21	4	5	1	38	4	0	38*	0	5	0.63
mggdb10	4	12	38	6	9	12	3	258	2,047	3,243	268*	0	51,192	813.59	8	11	3	181	341	33	191*	0	100	3.46
mggdb15	4	7	32	5	5	10	3	44	3	0	44*	0	1	0.72	6	9	3	37	12	0	37*	0	1	0.21
mggdb20	4	11	34	5	6	11	3	93	398	7	96*	0	5,386	75.81	6	10	3	92	450	362	94*	0	885	15.87
mggdb4	4	11	30	4	6	9	2	180	464	662	242*	0	2,548	27.59	6	9	2	205	677	220	238*	0	684	9.77
mggdb1	5	12	34	5	7	11	3	232	1,106	3,746	252*	0	46,022	601.02	6	10	3	248.33	956	549	279*	0	24,247	392.87
mggdb14	5	7	32	5	5	10	3	81	315	1	84*	0	39,743	369.34	6	9	3	58	293	1,515	62*	0	5,657	76
mggdb16	5	8	42	7	5	13	4	71.5	870	5,413	75*	0	138,513	2,271.37	5	12	4	84	8	0	84*	0	1	0.74
mggdb17	5	8	42	7	6	13	4	62	12	0	62*	0	1	0.85	5	12	4	65	20	0	65*	0	1	0.49
mggdb3	5	12	34	5	6	11	3	208	1,483	2,957	243*	0	104,446	1,700.95	7	10	3	208	338	0	225*	0	6,285	88.16
mggdb6	5	12	34	5	7	11	3	235	1,467	0	262*	0	95,001	1,552.37	6	10	3	247	822	569	270*	0	27,537	418.86
mggdb7	5	12	34	5	8	11	3	247	1,277	406	272*	0	24,435	412.65	6	10	3	215	2,400	24,387	282*	0	390,172	7,631.93
mggdb11	5	22	68	11	12	22	7	293	6,435	2,837	303*	0	44,282	3,875.68	12	20	6	270	8,792	75,753	283	2.6	139,844	–
mggdb18	5	9	54	9	8	17	5	135	71	0	135*	0	1	0.41	6	16	5	114	351	0	119*	0	1,442	45.3
mggdb21	6	11	50	8	7	16	5	117.5	1,973	55,179	120	2.07	485,914	–	9	15	4	100.25	954	140	104*	0	4,284	207.58
mggdb13	6	10	42	7	7	13	4	402	2,702	143,147	417	2.98	549,443	–	7	12	4	373	3,108	118,158	405	7.25	570,106	–
mggdb5	6	13	40	6	7	13	3	282	2,842	18,378	309	6.11	578,534	–	7	12	3	289	2,833	167,587	344	6.97	569,200	–
mggdb2	6	12	40	6	6	13	3	271	573	500	284*	0	79,016	1,435.16	7	12	3	281	1,635	46,745	308	1.86	809,124	–
mggdb12	7	13	36	5	6	11	3	369	2,859	27,195	461*	0	474,764	15,229.89	6	10	3	314	2,170	22,227	412*	0	378,525	10,608.79

Table 2 continued

File	m	$\|V\|$	$\|A\|$	$\|E\|$	$\beta = 0.35$										$\beta = 0.40$									
					$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME
mgval1A	2	24	35	20	12	22	13	158	61	0	158*	0	1	0.18	15	21	12	164	729	0	165*	0	5	1.34
mgval2A	2	24	28	16	12	18	10	280.5	399	22	286*	0	13	1.99	13	16	9	212	474	15	222*	0	21	1.7
mgval3A	2	24	33	15	13	21	9	81.67	1,385	20	84*	0	15	2.94	13	19	9	86	1,999	31	86*	0	1	0.4
mgval10A	3	50	106	32	34	68	20	472	47,039	2,103	475*	0	3,935	2,230.29	36	63	19	404	47,894	4,591	406*	0	8,734	5,107.91
mgval1B	3	24	38	13	16	24	8	188	1,703	649	192*	0	882	39.57	14	22	7	190	2,872	19,893	196*	0	31,292	1,414.98
mgval2B	3	24	40	12	13	26	7	305.67	6,184	1,004	326*	0	2,915	135.13	18	24	7	276	9,226	19,887	311*	0	39,334	1,505.98
mgval3B	3	24	29	16	13	18	10	105	496	1813	113*	0	2,840	128.98	14	17	9	107	224	22	110*	0	96	7.25
mgval4A	3	41	69	26	24	44	16	415	130,363	29,191	430	1.94	59,769	–	26	41	15	384	56,887	16,729	400*	0	54,305	11,152.63
mgval5A	3	34	74	22	20	48	14	436	116,626	52,082	454	3.3	78,100	–	25	44	13	423.33	17,122	964	426*	0	1,087	195.29
mgval6A	3	31	47	22	20	30	14	246	8,097	3	248*	0	303	50.23	20	28	13	220.5	8,742	1,132	224*	0	1,452	150.18
mgval7A	3	40	50	36	23	32	23	264	2,325	0	264*	0	1	1.19	25	30	21	269	14,524	0	271*	0	5,140	956.9
mgval8A	3	30	76	20	22	49	13	414	101,416	2	415	0.24	85,508	–	23	45	12	391	59,623	13,620	393*	0	24,031	4,331.64
mgval9A	3	50	100	32	31	65	20	322	59,102	1,288	324*	0	1,445	1,071.31	35	60	19	337	255,702	0	341	0.73	16,147	–
mgval10B	4	50	101	33	32	65	21	457	151,351	628	461	0.87	4,811	–	34	60	19	430.5	109,972	9,770	433	0.58	9,675	–
mgval4B	4	41	83	19	25	53	12	488	134,075	31,403	531	5.85	34,903	–	29	49	11	395	103,287	27,085	423	4.28	32,099	–
mgval5B	4	34	56	35	23	36	22	439	94,216	20,416	467	4.57	33,172	–	23	33	21	401.78	72,338	12,324	424	4.24	45,572	–
mgval6B	4	31	44	22	20	28	14	241	18,875	5,914	250*	0	10,514	1,715.66	19	26	13	203.22	19,802	1,907	211*	0	5,194	858.26
mgval7B	4	40	66	25	21	42	16	321	30,160	476	325*	0	181	208.79	23	39	15	258.68	32,968	2,776	270*	0	6,969	1,608.99
mgval8B	4	30	64	27	20	41	17	376	93,633	19,571	385	2.34	30,387	–	23	38	16	356	95,051	28,905	371	3.42	37,604	–
mgval9B	4	50	76	44	29	49	28	320.5	115,445	14,057	332	3.46	14,850	–	34	45	26	319	147,175	8,707	327	2.45	8,644	–
mgval10C	5	50	100	36	34	65	23	411	184,082	6,532	431	4.61	4,037	–	33	60	21	417	162,314	7,306	432	2.81	6,000	–
mgval4C	5	41	82	21	27	53	13	479.2	99,796	15,377	516	7.13	13,697	–	28	49	12	424	110,286	15,012	462	6.94	16,136	–
mgval5C	5	34	81	17	19	52	11	538.5	85,630	25,110	586	7.77	34,648	–	26	48	10	487.83	97,404	16,808	524	6.9	25,043	–
mgval9C	5	50	83	42	35	53	27	316	170,007	4,763	329	3.95	2,914	–	30	49	25	280	160,635	6,161	295	5.08	5,588	–
mgval3C	7	24	25	18	13	16	11	130	11,638	91,085	150	6.84	132,307	–	13	15	10	120	9,297	111,593	148	12.82	135,147	–

Table 3 Computational results for the datasets with $\beta \in \{0.45, 0.50\}$

File	m	$\|V\|$	$\|A\|$	$\|E\|$	$\beta = 0.45$										$\beta = 0.50$									
					$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME
mggdb19	3	8	18	2	3	4	1	40	39	3	48*	0	25	0.86	3	4	1	39.74	23	21	44*	0	5	0.34
mggdb10	4	12	38	6	9	10	3	196	1,045	259	214*	0	5,591	84.85	7	9	3	190	241	3	194*	0	118	3.71
mggdb15	4	7	32	5	6	8	2	34	20	0	34*	0	1	0.49	5	8	2	37	14	0	37*	0	1	0.42
mggdb20	4	11	34	5	5	9	2	76	187	80	78*	0	356	7.22	5	8	2	75	354	9	81*	0	1,458	20.7
mggdb4	4	11	30	4	7	8	2	186.33	273	111	228*	0	687	10.32	6	7	2	172	480	313	219*	0	1,441	16.42
mggdb1	5	12	34	5	6	9	2	211	812	4,514	259*	0	18,664	283.37	8	8	2	178	625	154	214*	0	5,381	86.77
mggdb14	5	7	32	5	6	8	2	58	306	216	66*	0	4,259	102.7	6	8	2	71	168	589	75*	0	1,903	20.55
mggdb16	5	8	42	7	6	11	3	62	762	1,696	70*	0	17,039	238.22	6	10	3	62	230	38	66*	0	2,367	36.26
mggdb17	5	8	42	7	7	11	3	53	83	0	53*	0	26	1.69	7	10	3	53	8	0	53*	0	1	0.92
mggdb3	5	12	34	5	8	9	2	210	837	852	237*	0	16,917	246.12	9	8	2	184	948	2,501	218*	0	23,111	399.66
mggdb6	5	12	34	5	7	9	2	182	372	21	218*	0	7,676	108.99	7	8	2	228	1,634	21,742	276*	0	73,338	1,171.76
mggdb7	5	12	34	5	9	9	2	171	1,837	3,456	243*	0	134,269	2,121.23	9	8	2	237	1,109	4,890	265*	0	24,566	378.97
mggdb11	5	22	68	11	15	18	6	278	11,895	58,465	297	4.41	136,575	–	16	17	5	249	10,157	21,229	275	4.16	151,520	–
mggdb18	5	9	54	9	7	14	4	112	4,419	7,269	123	7.19	765,445	–	8	13	4	111	3,461	24,431	121	6.63	830,877	–
mggdb21	6	11	50	8	7	13	4	120	2,498	90	122*	0	352,074	10,530.75	8	12	4	80	2,271	25,154	86*	0	452,227	15,691.34
mggdb13	6	10	42	7	7	11	3	371	2,933	176,047	423	10.19	636,018	–	8	10	3	214	1,894	10,885	259	8.52	602,727	–
mggdb5	6	13	40	6	7	11	3	309	2,827	76,185	350	4.43	639,825	–	7	10	3	226	1,772	41,212	292*	0	104,678	2,889.28
mggdb2	6	12	40	6	7	11	3	279	1,851	151,772	298	3.78	680,700	–	6	10	3	233	2,127	157,175	269	6.81	917,449	–
mggdb12	7	13	36	5	10	9	2	321	2,555	128,479	393	5.48	542,776	–	8	9	2	352	2,799	83,702	445*	0	392,073	13,492.66

Table 3 continued

File	m	$\|V\|$	$\|A\|$	$\|E\|$	$\beta = 0.45$										$\beta = 0.50$									
					$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	LB	CON	SUR	λ	GAP	NODES	TIME
mgval1A	2	24	35	20	17	19	11	166	2973	0	168*	0	107	7.11	16	17	10	137	2,832	28	145*	0	164	6.28
mgval2A	2	24	28	16	15	15	8	251	31	7	251*	0	1	0.49	16	14	8	248	569	4	248*	0	1	0.57
mgval3A	2	24	33	15	15	18	8	78	2,963	89	82*	0	80	4.8	17	16	7	71.09	2,939	59	75*	0	14	1.62
mgval10A	3	50	106	32	40	58	17	385	103,517	23,133	388	0.68	39,349	–	40	53	16	377.5	129,562	21,395	385	1.26	31,819	–
mgval1B	3	24	38	13	14	20	7	166	955	106	166*	0	8	3.2	17	19	6	170	732	120	170*	0	18	2.92
mgval2B	3	24	40	12	18	22	6	281	5,744	16,266	314*	0	32,816	1,311.99	18	20	6	262	3,741	4,491	284*	0	7,898	278.96
mgval3B	3	24	29	16	16	15	8	87	236	65	91*	0	28	3.23	15	14	8	98.5	899	15, 718	107*	0	27,612	801.35
mgval4A	3	41	69	26	29	37	14	365	58,282	3,700	381*	0	5,734	1,198.9	31	34	13	346	23,017	786	350*	0	810	144.85
mgval5A	3	34	74	22	26	40	12	377.38	70, 030	35,751	391	2.87	73,597	–	27	37	11	360	39,780	4,742	367*	0	7,957	1,296
mgval6A	3	31	47	22	23	25	12	207	12,407	0	213*	0	5,044	528.51	19	23	11	199	9,963	1,036	210*	0	1,324	123.21
mgval7A	3	40	50	36	27	27	19	257.5	32,491	56	261*	0	853	200.07	31	25	18	243	15,940	114	248*	0	600	126.78
mgval8A	3	30	76	20	24	41	11	367	22,765	64,078	370	0.6	110,057	–	26	38	10	382	71,760	11,678	388	1.35	151,496	–
mgval9A	3	50	100	32	37	55	17	299	252,748	17,103	306	2.11	23,947	–	39	50	16	306	21,342	119	306*	0	1	4.05
mgval10B	4	50	101	33	35	55	18	390	148,559	7,900	399	2.26	12,116	–	44	50	16	364	135,310	10	369	1.36	12,826	–
mgval4B	4	41	83	19	36	45	10	423	91,278	27,348	471	5.12	33,170	–	32	41	9	361	60,767	33,453	413	4.2	46,437	–
mgval5B	4	34	56	35	26	30	19	378.61	57,879	22,961	416	5.61	51,360	–	28	28	17	348	51,777	22,287	378	3.71	58,097	–
mgval6B	4	31	44	22	22	24	12	192.05	12,108	2,215	210*	0	6,682	892.69	23	22	11	192.5	18,566	1,246	210*	0	3,066	368.05
mgval7B	4	40	66	25	28	36	13	290	46,908	650	294	1.28	53,735	–	26	33	12	272	16,858	169	276*	0	9,935	1,429.5
mgval8B	4	30	64	27	23	35	14	341	79,601	28,522	360	4.68	48,260	–	23	32	13	330	92,237	34,868	350	4.69	58,830	–
mgval9B	4	50	76	44	35	41	24	311	124,190	11,830	323	3.36	15,284	–	37	38	22	261.65	121,647	16,409	278	3.54	20,704	–
mgval10C	5	50	100	36	37	55	19	382	221,922	9,405	403	3.83	8,715	–	40	50	18	389	122,174	7,937	406	4.19	6,086	–
mgval4C	5	41	82	21	29	45	11	434	94,762	18,519	481	9.18	18,753	–	32	41	10	441.5	96,970	23,184	488	9.1	23,040	–
mgval5C	5	34	81	17	26	44	9	445	57,530	25,167	492	7.84	39,574	–	26	40	8	416.5	60,718	21,691	459	8.41	43,299	–
mgval9C	5	50	83	42	34	45	23	273	148,877	7,430	291	5.88	8,710	–	38	41	21	278.33	137,567	5,727	301	7.44	4,980	–
mgval3C	7	24	25	18	16	13	9	122	7,489	129,628	143	8.7	161,002	–	15	12	9	116.25	6,552	129,396	137	7.94	184,169	–

Table 4 Computational results for the *mval* instances

FILE	m	$\|V\|$	$\|A\|$	$\|E\|$	$\|V_R\|$	$\|A_R\|$	$\|E_R\|$	UB	LB	CON	SUR	λ	GAP	NODES	TIME
mval1A	2	24	35	20	0	35	20	230	226	2,811	0	230	0	113	5.53
mval2A	2	24	28	16	0	28	16	324	319	1,812	0	324	0	3	0.45
mval3A	2	24	33	15	0	33	15	115	115	63	1	115	0	1	0.14
mval10A	3	50	106	32	0	106	32	634	634	18	0	634	0	1	5.97
mval1B	3	24	38	13	0	38	13	261	261	193	0	261	0	1	1.55
mval2B	3	24	40	12	0	40	12	395	384	16,529	1,644	395	0	36,724	1,813.66
mval3B	3	24	29	16	0	29	16	142	140	1,260	1,285	142	0	1,856	72.75
mval4A	3	41	69	26	0	69	26	580	576	67,713	1,204	580	0	1,423	591.60
mval5A	3	34	74	22	0	74	22	597	597	3,966	35	597	0	1	0.07
mval6A	3	31	47	22	0	47	22	326	326	2,777	0	326	0	1	0.06
mval7A	3	40	50	36	0	50	36	364	363	163,153	0	364	0	23,012	8,551.50
mval8A	3	30	76	20	0	76	20	581	581	1,742	0	581	0	1	3.10
mval9A	3	50	100	32	0	100	32	458	458	32,101	0	458	0	1	65.69
mval10B	4	50	101	33	0	101	33	661	658	221,856	0	661	0	1,719	5,763.42
mval4B	4	41	83	19	0	83	19	650	619.5	284,976	19,184	650	4.34	18,300	–
mval5B	4	34	56	35	0	56	35	613	599	206,158	6,251	613	2.28	21,965	–
mval6B	4	31	44	22	0	44	22	317	314	27,959	67	317	0	37,421	6,927.85
mval7B	4	40	66	25	0	66	25	412	411	24,630	0	412	0	27	60.11
mval8B	4	30	64	27	0	64	27	531	528	116,485	603	531	0	4,328	1,945.07
mval9B	4	50	76	44	0	76	44	453	453	49,192	122	453	0	1	108.82
mval10C	5	50	100	36	0	100	36	623	621	194,730	1,612	623	0	718	19,190.38
mval4C	5	41	82	21	0	82	21	630	616.5	206,072	12,823	630	2.14	10,848	–
mval5C	5	34	81	17	0	81	17	697	681	211,405	6,472	697	2.30	18,301	–
mval9C	5	50	83	42	0	83	42	429	428	118,825	401	428	0	1	1,552.23
mval3C	7	24	25	18	0	25	18	166	146	19,048	87,424	166	6.62	103,424	–

surrogate constraints is effective especially for the instances with a large number of vehicles.

Further experiments and final remarks. The performance of the B&C algorithm proposed in this paper has been evaluated by carrying out computational experiments on 12 datasets derived from classical datasets for the undirected and mixed CARP (our datasets for the MCGRP are available at ftp://160.97.54.1). The algorithm reaches the optimal solution in 154 of the 264 instances. In the instances that are not optimally solved, the average percentage gaps remain below a satisfactory threshold equal to 2.56, and bounds on the optimal solution value are provided. The study has confirmed that the complexity of the problem considerably increases whenever the number of vehicles rises. More effort is required to solve all the instances to optimality. Nevertheless, this work represents a first step to tackle the MCGRP directly by using formulation-based approaches.

Further experiments were carried out by considering *nearp* instances designed for the MCGRP [27] and limiting the computational investigation to those problems that require a small number of vehicles (4 problems). Specifically, we solve *nearp23* and *nearp12* to optimality. The optimal cost is equal to 780 and 3,138, respectively (780 and 3,235 are the upper bounds reported in [27], respectively). Our algorithm provides good upper bounds for instances *nearp22* and *nearp1*. Their costs are equal to 1,941 and 2,587, respectively (1,941 and 2,589 are the upper bounds reported in [27], respectively).

Obviously, our approach can be used to tackle pure NRP_s and ARP_s. For this reason, experiments were carried out also on *mval* instances, designed for the mixed CARP (instances with up to 7 vehicles). Bounds for such instances are provided in [3, 17]. The results obtained with our algorithm are shown in Table 4. The column headings have the same meaning as the headings in Tables 1–3, except for column UB that reports the upper bound values from [3]. The B&C algorithm always reaches the optimal value. More precisely, we observe that, for the instances where the GAP is more than 0, the objective value λ is equal to the best known lower bound. Finally, we underline the improvement on instance *mval9C*.

References

1. Belenguer, J.M., Benavent, E.: The capacitated arc routing problem: valid inequalities and facets. Comput. Optim. Appl. **10**(2), 165–187 (1998)
2. Belenguer, J.M., Benavent, E.: A cutting plane algorithm for the capacitated arc routing problem. Comput. Oper. Res. **30**(5), 705–728 (2003)
3. Belenguer, J.M., Benavent, E., Lacomme, P., Prins, C.: Lower and upper bounds for the mixed capacitated arc routing problem. Comput. Oper. Res. **33**(12), 3363–3383 (2006)
4. Benavent, E., Corberán, A., Sanchis, J.M.: Linear programming based methods for solving arc routing problems. In: Dror, M. (ed.) Arc Routing: Theory, Solutions and Applications, pp. 231–275. Kluwer Academic Publishers, Dordrecht (2000)
5. Corberán, A., Letchford, A.N., Sanchis, J.M.: A cutting plane algorithm for the general routing problem. Math. Program. Ser. A **90**(2), 291–316 (2001)
6. Corberán, A., Mejía, G., Sanchis, J.M.: New results on the mixed general routing problem. Oper. Res. **53**(2), 363–376 (2005)
7. Corberán, A., Plana, I., Sanchis, J.M.: A branch & cut algorithm for the windy general routing problem and special cases. Networks **49**(4), 245–257 (2007)
8. Corberán, A., Plana, I., Sanchis, J.M.: The windy general routing polyhedron: a global view of many known arc routing polyhedra. SIAM J. Discret. Math. **22**(2), 606–628 (2008)
9. Corberán, A., Prins, C.: Recent results on arc routing problems: an annotated bibliography. Networks **56**(1), 50–69 (2010)
10. Corberán, A., Romero, A., Sanchis, J.M.: The mixed general routing polyhedron. Math. Program. Ser. A **96**(1), 103–137 (2003)
11. Corberán, A., Sanchis, J.M.: The general routing problem polyhedron: facets from the RPP and GTSP polyhedra. Eur. J. Oper. Res. **108**(3), 538–550 (1998)
12. Dror, M. (ed.) Arc Routing: Theory, Solutions and Applications. Kluwer Academic Publishers, Dordrecht (2000)
13. Fischetti, M., Glover, F., Lodi, A.: The feasibility pump. Math. Program. **104**(1), 91–104 (2005)
14. Fischetti, M., Salazar, J.J., Toth, P.: A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Oper. Res. **45**(3), 378–394 (1997)
15. Golden, B.L., Dearmon, J.S., Baker, E.K.: Computational experiments with algorithms for a class of routing problems. Comput. Oper. Res. **10**(1), 47–59 (1983)
16. Golden B.L., Raghavan, S., Wasil, E.A. (eds.) The Vehicle Routing Problem: Latest Advances and New Challenges (Operations Research/Computer Science Interfaces Series). Springer, Berlin (2008)

17. Gouveia, L., Mourão, M.C., Pinto, L.S.: Lower bounds for the mixed capacitated arc routing problem. Comput. Oper. Res. **37**(4), 692–699 (2010)
18. Gutiérrez, J.C.A., Soler, D., Hervás, A.: The capacitated general routing problem on mixed graphs. Revista Investigacion Operacional **23**(1), 15–26 (2002)
19. Lenstra, J.K., Rinnooy Kan, A.H.G.: On general routing problems. Networks **6**(3), 273–280 (1976)
20. Letchford, A.N.: New inequalities for the general routing problem. Eur. J. Oper. Res. **96**(2), 317–322 (1997)
21. Letchford, A.N.: The general routing polyhedron: a unifying framework. Eur. J. Oper. Res. **112**(1), 122–133 (1999)
22. Orloff, C.S.: A fundamental problem in vehicle routing. Networks **4**(1), 35–64 (1974)
23. Padberg, M.W., Rao, M.R.: Odd minimum cut-sets and b-matchings. Math. Oper. Res. **7**(1), 67–80 (1982)
24. Pandit, R., Muralidharan, B.: A capacitated general routing problem on mixed networks. Comput. Oper. Res. **22**(5), 465–478 (1995)
25. Pearn, W.L., Assad, A., Golden, B.L.: Transforming arc routing into node routing problems. Comput. Oper. Res. **14**(4), 285–288 (1987)
26. Prim, R.C.: Shortest connection networks and some generalizations. Bell Syst. Tech. J. **36**, 1389–1401 (1957)
27. Prins, C., Bouchenoua, S.: A memetic algorithm solving the VRP, the CARP and general routing problems with nodes, edges and arcs. In: Hart, W.E., Krasnogor, N., Smith, J.E. (eds.) Recent Advances in Memetic Algorithms, pp. 65–85. Springer, Berlin (2005)
28. Santos, L., Coutinho-Rodrigues, J., Current, J.R.: An improved heuristic for the capacitated arc routing problem. Comput. Oper. Res. **36**(9), 2632–2637 (2009)