Math. Program., Ser. A (2008) 115:351–385
DOI 10.1007/s10107-007-0178-5

FULL LENGTH PAPER

An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts

Roberto Baldacci · **Nicos Christofides** · **Aristide Mingozzi**

Received: 1 August 2006 / Accepted: 26 June 2007 / Published online: 1 August 2007

Abstract This paper presents a new exact algorithm for the Capacitated Vehicle Routing Problem (CVRP) based on the set partitioning formulation with additional cuts that correspond to capacity and clique inequalities. The exact algorithm uses a bounding procedure that finds a near optimal dual solution of the LP-relaxation of the resulting mathematical formulation by combining three dual ascent heuristics. The first dual heuristic is based on the q-route relaxation of the set partitioning formulation of the CVRP. The second one combines Lagrangean relaxation, pricing and cut generation. The third attempts to close the duality gap left by the first two procedures using a classical pricing and cut generation technique. The final dual solution is used to generate a reduced problem containing only the routes whose reduced costs are smaller than the gap between an upper bound and the lower bound achieved. The resulting problem is solved by an integer programming solver. Computational results over the main instances from the literature show the effectiveness of the proposed algorithm.

Keywords Vehicle routing · Set partitioning · Dual ascent · Valid inequalities

Mathematics Subject Classification (2000) 90C27 · 49M29 · 90C39

R. Baldacci (✉)
DEIS, University of Bologna, Via Venezia 52, 47023 Cesena, Italy
e-mail: rbaldacci@deis.unibo.it

N. Christofides
Centre for Quantitative Finance, Imperial College, Exhibition Road, London SW7 2PG, UK
e-mail: n.christofides@imperial.ac.uk

A. Mingozzi
Department of Mathematics, University of Bologna, Via Sacchi 3,
47023 Cesena, Italy
e-mail: mingozzi@csr.unibo.it

1 Introduction

The Capacitated Vehicle Routing Problem (CVRP) considered in this paper is described as follows. An undirected graph $G = (V', E)$ is given where $V' = \{0, 1, \ldots, n\}$ is the set of $n + 1$ vertices and E is the set of edges. Vertex 0 represents the depot and the vertex set $V = V' \setminus \{0\}$ corresponds to n customers. A nonnegative cost d_{ij} is associated with each edge $\{i, j\} \in E$. We do not assume that the cost matrix $[d_{ij}]$ satisfies the triangle inequality. Each customer $i \in V$ requires a supply of q_i units from depot 0 (we assume $q_0 = 0$). A set of m identical vehicles of capacity Q is stationed at depot 0 and must be used to supply the customers. A *route* is defined as a least cost simple cycle of graph G passing through depot 0 and such that the total demand of the vertices visited does not exceed the vehicle capacity. The problem objective is to design m routes, one for each vehicle, so that all customers are visited exactly once and the sum of the route costs is minimized. The CVRP is $\mathcal{NP}$-hard as it is a natural generalization of the Travelling Salesman Problem (TSP).

Real-world CVRP's (see Ball et al. [8], Bodin et al. [11] and Toth and Vigo [43]) involve, in addition to vehicle capacity restrictions, complicated constraints like time windows to visit customers, customer-vehicle incompatibilities, mixed deliveries and collections on the same route, multiple interacting depots, etc.

The practical importance of the CVRP provides the motivation for the effort involved in the development of heuristic algorithms (see the surveys of Laporte and Semet [28] and of Gendreau et al. [23]). Surveys covering exact algorithms were given by Laporte [26] and Fisher [20]. The chapters of Toth and Vigo [42], Naddef and Rinaldi [38] and Bramel and Simchi-Levi [12] in the book edited by Toth and Vigo [43], survey the most effective exact methods proposed in the literature up to 2002. A recent survey of the CVRP, covering both exact and heuristic algorithms, can be found in the chapter of Cordeau et al. [16] of the book edited by Barnhart and Laporte [9].

The most promising exact algorithms for the symmetric CVRP which have been published since then are due to Baldacci et al. [6], Lysgaard et al. [32] and Fukasawa et al. [21].

Baldacci et al. [6] described a branch-and-cut algorithm that is based on a two-commodity network flow formulation of the CVRP. Lysgaard et al. [32] proposed a branch-and-cut algorithm that is an enhancement of the method proposed by Augerat et al. [2]. They used a variety of valid inequalities, including capacity, framed capacity, comb, partial multistar, hypotour and classical Gomory mixed integer cuts. The algorithms of Augerat et al. [2], of Baldacci et al. [6] and of Lysgaard et al. [32] were able to solve a 135-customer instance which is the largest non-trivial CVRP instance solved to date.

The best exact method currently available for the CVRP has been proposed by Fukasawa et al. [21]. This method combines the branch-and-cut of Lysgaard et al. [32] with the Set Partitioning (SP) approach. Besides the well-known capacity constraints, these authors also use framed capacity, strengthened comb, multistar, partial multistar, generalized multistar and hypotour inequalities, all presented in Lysgaard et al. [32]. The columns of the SP correspond to the set of q-routes that contains the set of valid CVRP routes. Since the resulting formulation has an exponential number of both columns and rows, this leads to column and cut generation for computing the lower bound and to a branch-and-cut-and-price algorithm for solving the CVRP. The computational results indicate that the new bounding procedure obtains lower bounds that are superior to those given by previous methods. However, this procedure is time consuming as the LP-relaxation of the master problem is usually highly degenerate and degeneracy implies alternative optimal dual solutions. Consequently, the generation of new columns and their associated variables may not change the value of the objective function of the master problem, the master problem may become large, and the overall method may become slow computationally. Moreover, in some CVRP instances, the increase in the lower bound with respect to the one achieved by the pure branch-and-cut method is very small and is not worth the computing time required by the additional SP approach. The exact algorithm presented by Fukasawa et al. [21] decides at the root node, according to the best balance between running time and bound quality, either to use the branch-and-cut method of Lysgaard et al. [32] or the new proposed branch-and-cut-and-price strategy. The computational results reported by Fukasawa et al. [21] show that this algorithm is very consistent on solving instances from the literature with up to 135 customers.

In this paper, we present an exact algorithm for solving the SP formulation of the CVRP that extends the exact methods for the CVRP proposed by Mingozzi et al. [36] and Baldacci et al. [4].

The method we propose in this paper avoids the drawbacks of the column and cut generation of Fukasawa et al. [21] as follows.

- We use an additive bounding procedure that combines three heuristics, called H^1, H^2 and H^3, to compute an effective dual solution of the LP-relaxation of the SP formulation improved by additional cuts. In H^1 and H^2 we add to the SP formulation capacity inequalities while in H^3 we add both capacity and clique inequalities. H^1 and H^2 are based on a relaxation of the SP formulation using variable splitting and relaxing in a Lagrangean fashion both partitioning and capacity constraints.
  H^1 is an extension of the bounding method proposed by Christofides et al. [15] and it is based on the q-route relaxation of the CVRP routes.
  H^2 is a column and cut generation procedure that considers valid CVRP routes, but solves the master problem using Lagrangean relaxation.
  H^3 is a column and cut generation procedure that attempts to close the duality gap of the near optimal dual solution produced by H^1 and H^2.
- The final dual solution achieved is used to generate a reduced SP problem containing only the routes whose reduced cost is smaller than the gap between an upper bound and the lower bound achieved. Then, the resulting reduced problem is solved by an integer programming solver.

Our computational results over the main instances from the literature show that this algorithm is competitive to both the exact methods of Lysgaard et al. [32] and of Fukasawa et al. [21].

This paper is organized as follows. In Sect. 2 we present the two-index and the SP formulations. In Sect. 3 we describe valid inequalities to improve the SP formulation. Section 4 describes the structure of the exact method and the additive procedure for computing the lower bound on the CVRP. Section 5 presents the three bounding procedures H^1, H^2 and H^3 used by the additive method. A dynamic programming algorithm for generating feasible routes is described in Sect. 6. Computational results on test instances taken from the literature are reported in Sect. 7. Finally, conclusions are given in Sect. 8.

2 Mathematical formulations

In this section we describe two classical formulations of the CVRP. The first formulation is the two-index vehicle flow formulation and the second one is the SP formulation.

In addition to the notation already introduced, for a subset $F \subseteq E$, $G(F)$ denotes the subgraph $(V'(F), F)$ induced by F, where $V'(F)$ is the set of vertices incident to at least one edge of F. Given a set $S \subseteq V$ let $\overline{S} = V' \backslash S$ be the *complement* of S and let $\delta(S)$ be the *cutset* defined by S (i.e., $\delta(S) = \{\{i, j\} \in E : i \in S, j \notin S$ or $i \notin S$, $j \in S\}$. Also, let $q(S) = \sum_{i \in S} q_i$ be the total demand of customers in S, and let $k(S)$ be the minimum number of vehicles of capacity Q needed to service all customers in S.

In the following we will use R and $E(R)$ to indicate the subset of vertices (i.e., $R = \{0, i_1, i_2, \ldots, i_h\}$) and the subset of edges of graph G visited by the route, respectively. Such a route represents the trip of one vehicle leaving the depot, delivering the demands of the customers in $R\backslash\{0\}$, and returning to the depot. The cost $c(R)$ of a route is equal to the cost of the solution to the TSP defined by the set R of vertices.

2.1 Two-index formulation

The two-index formulation of the CVRP was originally proposed by Laporte et al. [27] and is as follows.

Let x_{ij} be an integer variable which may take value $\{0, 1\}$, $\forall \{i, j\} \in E\backslash\{\{0, j\} : j \in V\}$ and value $\{0, 1, 2\}$, $\forall \{0, j\} \in E$, $j \in V$. Note that $x_{0j} = 2$ when a route including the single customer j is selected in the solution.

The CVRP can be formulated as the following integer program.

$$(F) \quad z(F) = \min \sum_{\{i,j\} \in E} d_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{\{i,j\} \in \delta(\{h\})} x_{ij} = 2, \quad \forall h \in V \tag{2}$$

$$\sum_{\{i,j\} \in \delta(S)} x_{ij} \geqslant 2k(S), \quad \forall S \in \mathscr{S} \tag{3}$$

$$\sum_{j \in V} x_{0j} = 2m \tag{4}$$

$$x_{ij} \in \{0, 1\}, \quad \forall \{i, j\} \in E\backslash\{\{0, j\} : j \in V\} \tag{5}$$

$$x_{0j} \in \{0, 1, 2\}, \quad \forall \{0, j\} \in E, j \in V, \tag{6}$$

where $\mathscr{S} = \{S : S \subseteq V, |S| \geqslant 2\}$.

Constraints (2) are the degree constraints for each customer. Constraints (3) are the *capacity constraints* (also called generalized subtour elimination constraints) which, for any subset S of customers that does not include the depot, impose that at least $k(S)$ vehicles enter and leave S. It is $\mathcal{NP}$-hard to compute $k(S)$ (see Naddef and Rinaldi [38]). However, Cornuéjols and Harche [17] have shown that formulation F remains valid if $k(S)$ is replaced by a lower bound such as $\lceil q(S)/Q \rceil$, where $\lceil x \rceil$ denotes the smallest integer not less than x. Constraint (4) states that m vehicles must leave and return to the depot while constraints (5) and (6) are the integrality constraints.

In the following, we denote by LF the LP-relaxation of formulation F.

2.2 Set Partitioning formulation

The SP formulation of the CVRP was originally proposed by Balinski and Quandt [7] and associates a binary variable with each feasible route. This model is valid for any type of cost matrix $[d_{ij}]$. However, if the cost matrix $[d_{ij}]$ satisfies the triangle inequality, then the SP formulation can be converted to a Set Covering formulation, preserving the optimal objective function value (see [12]).

Let $\mathcal{R}$ be the index set of all feasible routes and $\mathcal{R}_i \subset \mathcal{R}$ be the index subset of the routes covering customer $i \in V$. Let a_{ir} be a binary coefficient that is equal to 1 if vertex $i \in V'$ belongs to route $r \in \mathcal{R}$ and takes the value 0 otherwise (note that $a_{0r} = 1, \forall r \in \mathcal{R}$). Each route $r \in \mathcal{R}$ has an associated cost c_r. Let y_r be a (0–1) binary variable that is equal to 1 if and only if route $r \in \mathcal{R}$ belongs to the optimal solution.

The SP model of the CVRP is as follows:

$$(SP) \quad z(SP) = \min \sum_{r \in \mathcal{R}} c_r y_r \tag{7}$$

$$\text{s.t.} \sum_{r \in \mathcal{R}} a_{ir} y_r = 1, \quad \forall i \in V \tag{8}$$

$$\sum_{r \in \mathcal{R}} y_r = m \tag{9}$$

$$y_r \in \{0, 1\}, \quad \forall r \in \mathcal{R}. \tag{10}$$

Constraints (8) specify that each customer $i \in V$ must be covered by one route and, constraint (9) requires that m routes are selected.

Any SP solution $\mathbf{y}$ can be transformed into an F solution (see Baldacci et al. [6]) by setting:

$$x_{ij} = \sum_{r \in \mathcal{R}} \eta^r_{ij} y_r, \quad \forall \{i, j\} \in E, \tag{11}$$

where the coefficients η^r_{ij} are defined as follows:

- if r is a single customer route covering customer h, then $\eta^r_{0h} = 2$ and $\eta^r_{ij} = 0$, $\forall \{i, j\} \in E \backslash \{0, h\}$;

– if r is not a single customer route, then $\eta^r_{ij} = 1$ for each edge $\{i, j\} \in E(R_r)$ and $\eta^r_{ij} = 0, \forall \{i, j\} \in E \setminus E(R_r)$.

Problem SP has a possibly exponential number of variables and cannot be used directly to solve CVRP instances even of moderate size. However, the optimal solution cost $z(LSP)$ of the LP-relaxation of SP (called LSP) provides a very tight lower bound on the CVRP. Model SP is very general and can take into account several route constraints (e.g., time windows) because route feasibility is implicitly considered in the definition of set $\mathscr{R}$.

3 Set Partitioning formulation with additional cuts

The lower bound $z(LSP)$ can be strengthened by adding to LSP, in a cutting plane fashion, capacity constraints and all other inequalities known for the CVRP that are violated by the current LSP solution $\mathbf{y}$ such as comb, extended comb, framed capacity, hypotour, and so on (see Letchford et al. [29]). These inequalities are expressed in terms of variables x_{ij} defined in F but can be added to LSP once variables x_{ij} are replaced with variables y_r using Eq. (11). Moreover, relaxation LSP can be improved by adding any inequality valid for the Set Partitioning problem.

In this section we describe the valid inequalities we add to LSP and the corresponding separation problems.

3.1 Valid inequalities

(a) *Capacity constraints*
Using Eq. (11), constraint (3) become the following capacity inequalities for LSP:

$$\sum_{r \in \mathscr{R}(S)} \rho_r(S) y_r \geqslant 2k(S), \quad \forall S \in \mathscr{S}, \tag{12}$$

where $\mathscr{R}(S) = \{r \in \mathscr{R} : R_r \cap S \neq \emptyset\}$ and $\rho_r(S) = \sum_{\{i,j\} \in \delta(S)} \eta^r_{ij}$.
Inequalities (12) can be strengthened by observing (see Baldacci et al. [6]) that constraint (8) impose that in any feasible SP solution the number of routes covering the customers in S must be greater than or equal to $k(S)$. Therefore, stronger capacity inequalities than (12) are

$$\sum_{r \in \mathscr{R}(S)} y_r \geqslant k(S), \quad \forall S \in \mathscr{S}. \tag{13}$$

Inequalities (13) are a lifting of inequalities (12). It can be shown that a LSP solution $\mathbf{y}$ satisfying all constraints (12) can violate constraints (13) for some S as it is shown in the following example.
Let $\mathbf{y}$ be a LSP solution where five routes $R_1, R_2, \ldots, R_5$ have solution values $y_1 = y_2 = \cdots = y_5 = 0.5$. Consider the subset of customers

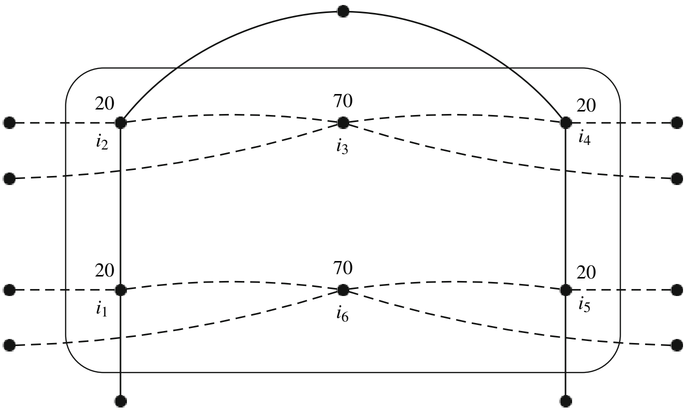

Fig. 1 Example of a set S that satisfies constraint (12) but violates constraint (13)

$S = \{i_1, i_2, i_3, i_4, i_5, i_6\}$. We have $k(S) = \lceil 220/100 \rceil = 3$ and $\rho_1(S) = 4$, $\rho_2(S) = \cdots = \rho_5(S) = 2$ and let $R_1 \cap S = \{i_1, i_2, i_4, i_5\}$, $R_2 \cap S = \{i_2, i_3\}$, $R_3 \cap S = \{i_1, i_6\}$, $R_4 \cap S = \{i_5, i_6\}$ and $R_5 \cap S = \{i_4, i_3\}$.
Customer demands are reported in Fig. 1 next to each customer. This solution satisfies constraint (12) for S since:

$$2y_1 + y_2 + y_3 + y_4 + y_5 = 3 = k(S),$$

but it violates the corresponding constraint (13) since:

$$y_1 + y_2 + y_3 + y_4 + y_5 = 2.5 \leqslant k(S).$$

Constraints (13) are separated as follows. We convert the LSP solution $\mathbf{y}$ into a LF solution $\mathbf{x}$ using Eq. (11) and we use the heuristic separation procedures of package CVRPSEP [31] to find any subset S whose constraint (3) is violated by $\mathbf{x}$. Then, for any such S, we add to LSP the corresponding constraint (13).

(b) *Other CVRP inequalities*
Any other CVRP inequalities designed for LF, such as comb, extended comb, framed capacity and hypotour that are violated by the LF solution $\mathbf{x}$ associated to the LSP solution $\mathbf{y}$ can be added to LSP to improve the lower bound.
The family $\mathscr{F}$ of these inequalities can be expressed in a general form for problem LF as:

$$\sum_{\{i,j\}\in E} \alpha^t_{ij} x_{ij} \geqslant \beta^t, \quad t \in \mathscr{F}. \tag{14}$$

Using Eq. (11), inequalities (14) become the following valid inequalities for LSP:

$$\sum_{r\in\mathscr{R}} \alpha^t(R_r) y_r \geqslant \beta^t, \quad t \in \mathscr{F}, \tag{15}$$

where $\alpha^t(R_r) = \sum_{\{i,j\}\in E(R_r)} \alpha^t_{ij}\eta^r_{ij}$.
These cuts are separated as follows. We convert the LSP solution $\mathbf{y}$ into a LF solution $\mathbf{x}$ and we use the CVRPSEP package [31] to find violated inequalities (14) and we add to LSP the corresponding inequalities (15).

(c) *Set Partitioning inequalities*
Relaxation LSP can be further improved by adding any inequality valid for the Set Partitioning problem (see Balas and Padberg [3] and Hoffman and Padberg [24]). However, in this paper we consider clique inequalities only.
Let $H = (\mathscr{R}, \mathscr{E})$ be the *conflict graph* where each node corresponds to a route and the edge set $\mathscr{E}$ contains every pair $\{r, r'\}, \forall r, r' \in \mathscr{R}, r < r'$, such that $R_r \cap R_{r'} \neq \{0\}$. We denote by $\mathscr{C}$ the set of all cliques of H. Clique inequalities are:

$$\sum_{r\in C} y_r \leqslant 1, \;\; C \in \mathscr{C}. \tag{16}$$

Let $H(\mathbf{y}) = (\mathscr{R}(\mathbf{y}), \mathscr{E}(\mathbf{y}))$ be the subgraph of graph H induced by the LSP solution $\mathbf{y}$ where $\mathscr{R}(\mathbf{y}) = \{r \in \mathscr{R} : y_r > 0\}$, and let y_r be the weight of vertex $r \in \mathscr{R}(\mathbf{y})$.
The separation problem associated with inequalities (16) is strongly $\mathscr{N}\mathscr{P}$-hard as it corresponds to find the maximal weighted clique of graph $H(\mathbf{y})$ (see [22]). Nevertheless, because the dimensions of graph $H(\mathbf{y})$ (i.e., the cardinality of the sets $\mathscr{R}(\mathbf{y})$ and $\mathscr{E}(\mathbf{y})$) are in our case small, we found to be computationally convenient to use the CLIQUER 1.1 package [39]. The CLIQUER package is composed of a set of C routines for finding cliques in an arbitrary weighted graph based on the exact branch-and-bound algorithm developed by Östergård [40]. In particular, the separation of clique inequalities is performed as follows. First, using the CLIQUER function *clique_ find_all(.)*, we look in graph $H(\mathbf{y})$ for all maximal cliques with weight at least min_w and at most max_w. If no such cliques are found, then we compute the maximal weighted clique of graph $H(\mathbf{y})$, again using CLIQUER function *clique_ find_all(.)*, but called using different input parameters.
After having performed a number of preliminary computational experimentations using different values of min_w and of max_w, for obtaining the computational results reported in Sect. 7 we decided to set $min_w = 1.3$ and $max_w = 4.0$.

3.2 Improved formulation for the CVRP

We denote by RP the integer problem that derives from SP adding inequalities (13), (15) and (16). Problem RP is as follows:

$$(RP) \quad z(RP) = \min \sum_{r\in\mathscr{R}} c_r y_r \tag{17}$$

$$\text{s.t.} \sum_{r\in\mathscr{R}} a_{ir} y_r = 1, \quad \forall i \in V \tag{18}$$

$$\sum_{r\in\mathscr{R}} y_r = m \tag{19}$$

$$\sum_{r\in\mathscr{R}(S)} y_r \geqslant k(S), \quad \forall S \in \mathscr{S} \tag{20}$$

$$\sum_{r\in\mathscr{R}} \alpha^t(R_r) y_r \geqslant \beta^t, \quad \forall t \in \mathscr{F} \tag{21}$$

$$\sum_{r\in C} y_r \leqslant 1, \quad \forall C \in \mathscr{C} \tag{22}$$

$$y_r \in \{0,1\}, \quad \forall r \in \mathscr{R}. \tag{23}$$

In the following section we describe a bounding procedure for computing a lower bound on the CVRP that consists of finding a near-optimal solution of the dual of the LP-relaxation of problem RP, called problem DRP.

Let $u = (u_0, u_1, \ldots, u_n)$ be a vector of dual variables, where u_i, $i \in V$, and u_0 are associated with constraints (18) and (19), respectively. Moreover, let v_S, $S \in \mathscr{S}$, w_t, $t \in \mathscr{F}$, and g_C, $C \in \mathscr{C}$, be the dual variables of constraints (20), (21) and (22), respectively. The dual problem DRP is as follows:

$$(DRP) \quad z(DRP) = \max \sum_{i\in V} u_i + m u_0 + \sum_{S\in\mathscr{S}} k(S) v_S + \sum_{t\in\mathscr{F}} \beta^t w_t + \sum_{C\in\mathscr{C}} g_C \tag{24}$$

$$\text{s.t.} \sum_{i\in V'} a_{ir} u_i + \sum_{S\in\mathscr{S}} b_r(S) v_S + \sum_{t\in\mathscr{F}} \alpha^t(R_r) w_t + \sum_{C\in\mathscr{C}_r} g_C \leqslant c_r, \quad \forall r \in \mathscr{R} \tag{25}$$

$$u_i \in \mathbb{R}, \quad \forall i \in V' \tag{26}$$

$$v_S \geqslant 0, \quad \forall S \in \mathscr{S} \tag{27}$$

$$w_t \geqslant 0, \quad \forall t \in \mathscr{F} \tag{28}$$

$$g_C \leqslant 0, \quad \forall C \in \mathscr{C}, \tag{29}$$

where $\mathscr{C}_r = \{C \in \mathscr{C} : r \in C\}$ and the coefficient $b_r(S)$ is equal to 1, $\forall r \in \mathscr{R}$, such that $R_r \cap S \neq \emptyset$ and $b_r(S) = 0$ otherwise.

Both problem RP and DRP are impractical to solve, even for moderate size CVRP instances since the numbers of variables and constraints of both problems are typically exponential. In Sect. 5 we describe an effective method for solving DRP that uses in sequence three heuristic procedures each one exploiting a different structure of the problem and without requiring the a priori generation of the entire route set $\mathscr{R}$.

4 An exact method for solving the CVRP

In this section we describe an exact algorithm for solving the RP formulation of the CVRP. The procedure generalizes the exact method proposed by Mingozzi, Christofides and Hadjiconstantinou [36] and by Baldacci et al. [4].

4.1 Description of the exact method

The core of the exact method proposed in this paper is the bounding procedure HDRP that uses in sequence three heuristics H^1, H^2 and H^3 to obtain a near optimal DRP solution $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ of cost $z'(DRP)$ without generating all routes and constraints (20), (21) and (22).

The exact method consist of finding, by means of an integer programming solver (such as CPLEX [18]), an optimal integer solution of problem RP resulting from the following reductions:

1. We replace the route set $\mathscr{R}$ with the subset $\mathscr{R}' \subset \mathscr{R}$ containing all routes whose reduced cost, with respect to the dual variables $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ achieved by HDRP is smaller that the gap $z_{UB} - z'(DRP)$, where z_{UB} is a valid upper bound on the CVRP. The route set $\mathscr{R}'$ is generated by procedure GENROUTE as described in Sect. 6.
2. We ignore constraints (21) and we consider only those constraints (20) and (22) generated by HDRP that are saturated by the final fractional RP solution obtained by H^3.

The effectiveness of the proposed exact method is based on having an effective procedure for generating a near optimal dual solution $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$. As the dual solution gets better, the reduced costs of the routes of an optimal CVRP solution get smaller and, hopefully, the size of subset $\mathscr{R}'$ that must be generated to find an optimal solution gets smaller.

The method HDRP for computing the DRP solution is given in the following section.

4.2 Bounding procedure HDRP

HDRP is an additive bounding method that computes a lower bound on the $CVRP$ as the sum of the solution costs obtained by heuristics H^1, H^2 and H^3 for solving DRP. HDRP is based on the additive method of Fischetti and Toth [19] for combinatorial optimization problems, but it extends it combining the LP-relaxation of alternative mathematical formulations.

Procedure HDRP was introduced by Bianco et al. [10] and used by Mingozzi et al. [35], Mingozzi et al. [37], Baldacci et al. [5] and Baldacci et al. [4]. Procedure HDRP is now summarized.

Let H^1, H^2, H^3 be three different bounding procedures for the CVRP such that each procedure H^t, $t = 1, 2, 3$, produces a lower bound LB_t on the CVRP corresponding to the cost of a feasible DRP solution, $(\mathbf{u}^t, \mathbf{v}^t, \mathbf{w}^t, \mathbf{g}^t)$. HDRP is an iterative bounding procedure that applies, at each iteration t, the heuristic procedure H^t to the residual problem DRP' that is derived from DRP by replacing the route cost c_r with the reduced cost c'_r, $r \in \mathscr{R}$, relative to the DRP solution $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ of cost $z'(DRP)$ found at the end of the previous iteration. At the end of iteration t, we have $\mathbf{u}' = \mathbf{u}' + \mathbf{u}^t$, $\mathbf{v}' = \mathbf{v}' + \mathbf{v}^t$, $\mathbf{w}' = \mathbf{w}' + \mathbf{w}^t$, $\mathbf{g}' = \mathbf{g}' + \mathbf{g}^t$ and $z'(DRP) = z'(DRP) + LB_t$.

At the first iteration, DRP' corresponds to DRP since $\mathbf{u}' = \mathbf{0}, \mathbf{v}' = \mathbf{0}, \mathbf{w}' = \mathbf{0}, \mathbf{g}' = \mathbf{0}$. It can be shown that the final lower bound $z'(DRP)$ is greater than or equal to the maximum of the lower bounds produced by each individual procedure when applied to the original problem DRP. In the following we use RP' to indicate the problem resulting from RP by replacing route costs c_r with the reduced costs c'_r so that at each iteration of HDRP, problem DRP' represents the dual of the LP-relaxation of RP'.

5 Bounding procedures for $CVRP$

We now describe the bounding procedures H^1, H^2 and H^3 used by HDRP for computing a lower bound on the CVRP.

Procedures H^1 and H^2 are based on a Lagrangean relaxation of problem RP and do not require the generation of the entire route set $\mathscr{R}$. H^1 is based on the q-route relaxation of the CVRP and extends the method proposed by Christofides et al. [15]. H^2 combines the Lagrangean relaxation with column generation to avoid the a priori generation of the route set $\mathscr{R}$ and cut generation to separate violated inequalities. This procedure requires much less computing time than using simplex based LP solvers to solve the master problem since it is not affected by the typical degeneration of the latter methods. Procedure H^3 is a column and cut generation procedure where the master is solved by a simplex based method.

These procedures can be used by HDRP in any order, since the correctness of the final DRP solution does not require any particular ordering of the component procedures. Our computational experiments indicated that the best lower bound was obtained by applying H^1 first, then H^2 and then H^3. Procedure H^3 is time consuming if it is the first procedure used. But when it is used as the last procedure in the sequence H^3, requires few iterations to close the duality gap left by the near optimal solution produced by H^1 and H^2.

5.1 Lagrangean problem LRP

Both procedures H^1 and H^2 do not require the a priori generation of the route set $\mathscr{R}$ and are based on the Lagrangean problem LRP that is obtained by relaxing RP as follows:

(a) We ignore constraints (21) and (22).
(b) We consider only a limited subset $\overline{\mathscr{S}} \subset \mathscr{S}$ of the capacity constraints that is generated as follows. We compute a lower bound to CVRP by solving problem LF adding, in a cutting plane fashion, constraints (3). These constraints are separated using the package CVRPSEP [31]. The set $\overline{\mathscr{S}}$ corresponds to the final set of constraints (3) found by this procedure.
(c) In order to make the relaxed problem LRP easier to solve, we split each variable y_r by $|R_r| - 1$ variables $\xi_r^i \in \{0, 1\}$, $i \in R_r \backslash \{0\}$, where ξ_r^i is equal to 1 if and only if customer i is visited by route r. The way to split variables y_r, $r \in \mathscr{R}$, is according to the following expressions:

$$y_r = \sum_{i \in V} a_{ir} \frac{q_i}{q(R_r)} \xi_r^i, \quad r \in \mathscr{R}. \tag{30}$$

where $q(R_r) = \sum_{i\in V} a_{ir} q_i$. Expression (30) imposes that if $y_r = 1$ then $\xi_r^i = 1$, $i \in R_r$, while if $y_r = 0$ then $\xi_r^i = 0$, $i \in R_r$. Moreover, we add the constraints $\sum_{r\in\mathscr{R}_i} \xi_r^i = 1$, $i \in V$, to specify that each customer $i \in V$ must be covered by one route. It is clear that these constraints are redundant as they are equivalent to constraints (18). The problem resulting from relaxations (a) and (b) now becomes:

$$(RP') \quad z(RP') = \min \sum_{r\in\mathscr{R}} c_r \left(\sum_{i\in V} a_{ir} \frac{q_i}{q(R_r)} \xi_r^i \right) \tag{31}$$

$$\text{s.t.} \sum_{r\in\mathscr{R}} a_{ir} \left(\sum_{j\in V} a_{jr} \frac{q_j}{q(R_r)} \xi_r^j \right) = 1, \quad \forall i \in V \tag{32}$$

$$\sum_{r\in\mathscr{R}} \left(\sum_{i\in V} a_{ir} \frac{q_i}{q(R_r)} \xi_r^i \right) = m \tag{33}$$

$$\sum_{r\in\mathscr{R}(S)} \left(\sum_{i\in V} a_{ir} \frac{q_i}{q(R_r)} \xi_r^i \right) \geqslant k(S), \quad \forall S \in \overline{\mathscr{S}} \tag{34}$$

$$\sum_{r\in\mathscr{R}_i} \xi_r^i = 1, \quad \forall i \in V \tag{35}$$

$$\xi_r^i \in \{0, 1\}, \quad \forall i \in V, \ \forall r \in \mathscr{R}. \tag{36}$$

(d) We further relax problem RP' dualizing constraints (32), (33) and (34). We associate penalties $\lambda_i \in \mathbb{R}$, $i \in V$, with constraints (32), $\lambda_0 \in \mathbb{R}$ with constraint (33) and $\sigma_S \geqslant 0$, $S \in \overline{\mathscr{S}}$, with constraints (34).

Relaxations (a), (b) and (c) provide the following Lagrangean problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$.

$$LRP(\boldsymbol{\lambda},\boldsymbol{\sigma}) \quad z(LRP(\boldsymbol{\lambda},\boldsymbol{\sigma})) = \min \sum_{i\in V} \left(\sum_{r\in\mathscr{R}_i} (c_r - \lambda(R_r) - \sigma(R_r)) \frac{q_i}{q(R_r)} \xi_r^i + \lambda_i \right) + m\lambda_0 + \sum_{S\in\overline{\mathscr{S}}} k(S)\sigma_S \tag{37}$$

$$\text{s.t.} \sum_{r\in\mathscr{R}_i} \xi_r^i = 1, \quad \forall i \in V \tag{38}$$

$$\xi_r^i \in \{0, 1\}, \quad \forall i \in V, \forall r \in \mathscr{R}. \tag{39}$$

where $\lambda(R_r) = \sum_{i\in V} a_{ir}\lambda_i$ and $\sigma(R_r) = \sum_{S\in\overline{\mathscr{S}}} b_r(S)\sigma_S$.

In consequence of the variable splitting induced by Eq. (30), problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is now decomposable into n subproblems, one for each customer $i \in V$, that can be easily solved by inspection. Moreover, any optimal $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ solution can be transformed into a DRP solution of cost equal to $z(LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma}))$ as it will be shown in the following by Theorem 1. Problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ can be solved as follows.

Let $r_i \in \mathscr{R}_i$ be the index of the route visiting customer $i \in V$ such that

$$q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i}) = q_i \min_{r \in \mathscr{R}_i}\{(c_r - \lambda(R_r) - \sigma(R_r))/q(R_r)\}. \quad (40)$$

An optimal solution $\boldsymbol{\xi}$ of problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is given by:

$$\xi^i_{r_i} = 1 \quad \text{and} \quad \xi^i_r = 0, \quad \forall r \in \mathscr{R}_i \backslash \{r_i\}, \ i \in V. \quad (41)$$

The cost $z(LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma}))$ of the optimal solution $\boldsymbol{\xi}$ of problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ given by expressions (40) and (41) is:

$$\begin{aligned} z(LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})) = &\sum_{i \in V}(q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i}) + \lambda_i) \\ &+ m\lambda_0 + \sum_{S \in \overline{\mathscr{S}}} k(S)\sigma_S. \end{aligned} \quad (42)$$

The following Theorem 1 shows that any optimal solution of problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ also provides a feasible, but not necessarily optimal, solution of problem DRP.

Theorem 1 *Any optimal $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ solution $\boldsymbol{\xi}$ provides a feasible solution* $(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{g})$ *of the dual problem DRP of cost $z(DRP) = z(LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma}))$ by setting $\mathbf{w} = \mathbf{0}$, $\mathbf{g} = \mathbf{0}$ and by computing $\mathbf{u}$ and $\mathbf{v}$ according to the following expressions*:

$$\left.\begin{array}{lr} u_i = q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i}) + \lambda_i, \quad i \in V, \quad \textit{and} \quad u_0 = \lambda_0 & \text{(a)} \\ v_S = \sigma_S, \, S \in \overline{\mathscr{S}} & \text{(b)} \end{array}\right\} \quad (43)$$

where r_i is the index of the route in $\mathscr{R}_i$ that satisfies equality (40) *for customer $i \in V$* (*that is* $\xi^i_{r_i} = 1$).

Proof Let us consider the dual constraint (25) corresponding to route r. As $r \in \mathscr{R}_i$, $\forall i \in R_r$, then the following inequalities hold:

$$q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i}) \leqslant q_i(c_r - \lambda(R_r) - \sigma(R_r))/q(R_r), \quad \forall i \in R_r. \quad (44)$$

From expressions (43a) and (44) we obtain

$$u_i \leqslant q_i(c_r - \lambda(R_r) - \sigma(R_r))/q(R_r) + \lambda_i, \quad \forall i \in R_r, \quad (45)$$

and from inequalities (45) we derive:

$$\sum_{i \in V} a_{ir} u_i \leqslant \sum_{i \in V} a_{ir} q_i(c_r - \lambda(R_r) - \sigma(R_r))/q(R_r) + \sum_{i \in V} a_{ir}\lambda_i = c_r - \sigma(R_r). \quad (46)$$

Using the definition of $\sigma(R_r)$ and expressions (43b), (46) becomes:

$$\sum_{i \in V} a_{ir} u_i + \sum_{S \in \overline{\mathscr{S}}} b_r(S) v_S \leqslant c_r, \tag{47}$$

and, since $\mathbf{w} = \mathbf{0}$ and $\mathbf{g} = \mathbf{0}$, inequality (47) corresponds to the dual constraint (25) of route r. It is quite simple to show that the cost $z(DRP)$ of the DRP solution given by expressions (43) is equal to $z(LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma}))$. □

A valid lower bound on the CVRP can be achieved by solving the following Lagrangean dual problem:

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\sigma}} \{z(LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})\}. \tag{48}$$

Problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ cannot be solved directly as it involves the entire route set $\mathscr{R}$. In the following Sects. 5.2 and 5.3, we describe two iterative procedures, called H^1 and H^2, for solving problem (48), that do not require the a priori generation of the route set $\mathscr{R}$. Both H^1 and H^2 produce two feasible DRP solutions using Theorem 1.

The first procedure H^1, solves $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ by relaxing the requirement that a route is a simple cycle of graph G. The second procedure H^2, finds a $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ solution $\boldsymbol{\xi}$ replacing $\mathscr{R}$ with a limited subset $\overline{\mathscr{R}} \subset \mathscr{R}$ and uses column generation to update $\overline{\mathscr{R}}$.

5.2 Bounding procedure H^1

In this section we describe the bounding procedure H^1, that was originally proposed by Christofides et al. [15].

Let us consider the relaxed problem $LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ that derives from problem $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ by relaxing the requirement that a route is a simple cycle of graph G. This relaxation allows us to compute in pseudo-polynomial time a least cost (not necessarily simple) cycle passing through each customer $i \in V$ as follows.

Given the penalty vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$, the following *modified edge cost* is associated with each edge $\{i, j\}$ of graph G:

$$\overline{d}_{ij} = d_{ij} - \frac{1}{2}\lambda_i - \frac{1}{2}\lambda_j - \frac{1}{2} \sum_{S \in \overline{\mathscr{S}}(i,j)} \sigma_S, \tag{49}$$

where $\overline{\mathscr{S}}(i, j) = \{S \in \overline{\mathscr{S}} : \{i, j\} \in \delta(S)\}$. Using cost $\overline{d}_{ij}$, the modified cost $\overline{c}_r$ of a route R_r is

$$\overline{c}_r = \sum_{\{i,j\} \in E(R_r)} \overline{d}_{ij} = c_r - \lambda(R_r) - \sum_{S \in \overline{\mathscr{S}}} \left(\frac{1}{2} \sum_{\{i,j\} \in \delta(S)} \eta^r_{ij} \right) \sigma_S \tag{50}$$

As $b_r(S) \leqslant \frac{1}{2}\sum_{\{i,j\}\in\delta(S)} \eta^r_{ij}$ and $\sigma_S \geqslant 0, \forall S \in \overline{\mathscr{S}}$, we have:

$$\sigma(R_r) = \sum_{S\in\overline{\mathscr{S}}} b_r(S)\sigma_S \leqslant \sum_{S\in\overline{\mathscr{S}}} \left(\frac{1}{2}\sum_{\{i,j\}\in\delta(S)} \eta^r_{ij}\right)\sigma_S. \tag{51}$$

Therefore, from expressions (50) and (51), we have:

$$\overline{c}_r \leqslant c_r - \lambda(R_r) - \sigma(R_r), \quad \forall r \in \mathscr{R}. \tag{52}$$

Using the modified edge costs, define $\phi(i,q)$ to be the cost of a least cost (not necessarily simple) cycle $C(i)$ of G that passes through the depot and vertex $i \in V$ where the total demand of the customers visited by cycle $C(i)$ is equal to q ($q_i \leqslant q \leqslant Q$). $\phi(i,q)$ provides a lower bound to the cost $\overline{c}_r$ of any route $r \in \mathscr{R}_i$ such that $q(R_r) = q, \forall i \in V$. Because of inequalities (52), it is easy to see that:

$$h(i) = \min_{q_i\leqslant q\leqslant Q}\{q_i\phi(i,q)/q\} \leqslant q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i}), \quad i \in V, \tag{53}$$

where r_i is the route giving the minimum in expression (40). Therefore, $h(i) + \lambda_i$ is a valid lower bound to the value of dual variable u_i, $i \in V$, given by expressions (43).

Moreover, a valid lower bound $z(LRP'(\boldsymbol{\lambda},\boldsymbol{\sigma}))$ on the CVRP is given by:

$$z(LRP'(\boldsymbol{\lambda},\boldsymbol{\sigma})) = \sum_{i\in V}(h(i) + \lambda(i)) + m\lambda_0 + \sum_{S\in\overline{\mathscr{S}}} q(S)\sigma_S. \tag{54}$$

Also, it can be shown that Theorem 1 remain valid if in expression (43) each term $q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i})$ is replaced with $h(i)$.

The values $\phi(i,q)$, $\forall i \in V$ and $q_i \leqslant q \leqslant Q$ can be computed according to the method proposed by Christofides et al. [15].

Computing lower bound LB_1

H^1 is an iterative procedure that uses subgradient optimization to find a suboptimal solution to the following Lagrangean dual problem:

$$LB_1 = \max_{\boldsymbol{\lambda},\boldsymbol{\sigma}}\{z(LRP'(\boldsymbol{\lambda},\boldsymbol{\sigma}))\}. \tag{55}$$

H^1 starts by setting $\boldsymbol{\lambda} = \mathbf{0}$ and $\boldsymbol{\sigma} = \mathbf{0}$. At each iteration the procedure performs the following operations.

(i) For the current values of $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$, compute the costs $h(i)$, $i \in V$, using the modified edge costs $\overline{d}_{ij}$ and a solution $\boldsymbol{\xi}$ of the relaxed problem $LRP'(\boldsymbol{\lambda},\boldsymbol{\sigma})$ as follows. Let $\widetilde{\mathscr{R}}$ be the index set of the distinct cycles $C(i)$ corresponding to $h(i)$, $i \in V$, and let r_i be the index of the cycle in $\widetilde{\mathscr{R}}$ of cost $h(i)$, $i \in V$. The solution $\boldsymbol{\xi}$ of problem $LRP'(\boldsymbol{\lambda},\boldsymbol{\sigma})$ is defined, according to expressions (41), by setting $\xi^i_{r_i} = 1$ and $\xi^i_r = 0, \forall r \in \widetilde{\mathscr{R}}\backslash\{r_i\}, i \in V$.

(ii) The penalty vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$ are modified according to the subgradient method as follows. Associate with solution $\boldsymbol{\xi}$ the not necessarily feasible RP solution $\mathbf{y}$ using Eq. (30). Note that because the cycles in $\widetilde{\mathscr{R}}$ can be non-simple, then the coefficient a_{ir} represents the number of times vertex $i \in V$ is visited by cycle $r \in \widetilde{\mathscr{R}}$, and a_{ir} can be greater than one. Let ϑ_i, $i \in V$, and ϑ_0 denote the values of the left-hand-side of constraints (18) and (19) with respect to solution $\mathbf{y}$, respectively. Moreover, let $\theta(S)$, $S \in \overline{\mathscr{S}}$, be the values of left-hand-side of constraint (20). At each iteration of H^1, the values of $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$ are modified as follows.

$$\left.\begin{aligned}
&\lambda_i = \lambda_i - \epsilon\gamma(\vartheta_i - 1), \quad i \in V\\
&\lambda_0 = \lambda_0 - \epsilon\gamma(\vartheta_0 - m)\\
&\sigma(S) = \max\{0, \sigma(S) - \epsilon\gamma(\theta(S) - k(S))\}, \quad \forall S \in \overline{\mathscr{S}}
\end{aligned}\right\} \tag{56}$$

where ϵ is a positive constant and

$$\gamma = \frac{z_{UB} - z(LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma}))}{\sum_{i\in V}(\vartheta_i - 1) + (\vartheta_0 - m) + \sum_{S\in\overline{\mathscr{S}}}(\theta(S) - k(S))}. \tag{57}$$

Let $\boldsymbol{\lambda}^*$, $\boldsymbol{\sigma}^*$ and $h^*(i)$, $i \in V$, be the values that produce the best lower bound LB_1 (i.e., $LB_1 = z(LRP'(\boldsymbol{\lambda}^*, \boldsymbol{\sigma}^*))$ where these values are computed in $Maxt1$ iterations by procedure H^1. Let $\mathbf{u}^1$, $\mathbf{v}^1$, $\mathbf{w}^1$ and $\mathbf{g}^1$ denote the feasible DRP solution of cost $z(DRP) = LB_1$ given by expressions (43) using $h^*(i)$ instead of $q_i(c_{r_i} - \lambda(R_{r_i}) - \sigma(R_{r_i}))/q(R_{r_i})$, $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ and $\boldsymbol{\sigma} = \boldsymbol{\sigma}^*$. We have $u_i^1 = h^*(i) + \lambda_i^*$, $i \in V$, $u_0^1 = \lambda_0^*$, $v_S^1 = \sigma_S^*$, $S \in \overline{\mathscr{S}}$, $\mathbf{w}^1 = \mathbf{0}$ and $\mathbf{g^1} = \mathbf{0}$.

At the end of the first iteration of HDRP, we have an evaluation of the final DRP solution as $z'(DRP) = LB_1$ and dual variables $\mathbf{u}' = \mathbf{u}^1$, $\mathbf{v}' = \mathbf{v}^1$, $\mathbf{w}' = \mathbf{0}$, $\mathbf{g}' = \mathbf{0}$ as a solution to DRP.

Dynamic generation of $\overline{\mathscr{S}}$

In alternative to the a priori generation of the set $\overline{\mathscr{S}}$ (see point (b) of Sect. 5.1), we experimented a dynamic generation of $\overline{\mathscr{S}}$ that consists of setting $\overline{\mathscr{S}} = \emptyset$ at the beginning of H^1 and in updating, at each iteration of H^1, the set $\overline{\mathscr{S}}$ by separating the capacity constraints (12) violated by the current $LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ solution $\boldsymbol{\xi}$ as follows. Associate with the solution $\boldsymbol{\xi}$ a not necessarily feasible F solution $\mathbf{x}$ as follows. First, transform solution $\boldsymbol{\xi}$ into a (not necessarily feasible) RP solution $\mathbf{y}$ by means of expressions (30). Then, transform solution $\mathbf{y}$ into a not necessarily feasible F solution $\mathbf{x}$ using Eq. (11) where, in order to take into account the fact that the cycles in $\widetilde{\mathscr{R}}$ can be non-simple, the coefficient η_{ij}^r indicates the number of times edge $\{i, j\}$ is traversed by cycle $r \in \widetilde{\mathscr{R}}$. However, our computational experiments indicate that this method is time consuming and it does not provide a better lower bound than generating a priori $\overline{\mathscr{S}}$.

5.3 Bounding procedure H^2

Problem DRP' is the dual problem of RP' derived by replacing the route cost c_r with the reduced costs $c'_r = c_r - \sum_{i \in R_r} u'_i - \sum_{S \in \overline{\mathscr{S}}} b_r(S) v'_S$, $r \in \mathscr{R}$. These reduced costs are computed relative to the current DRP solution $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ achieved by procedure H^1.

Procedure H^2 is an iterative heuristic algorithm that uses subgradient optimization and column and cut generation to solve $LB_2 = \max_{\boldsymbol{\lambda}, \boldsymbol{\sigma}} \{z(LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma}))\}$, where $LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ is the Lagrangean relaxation of problem RP' described above (i.e., $LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ derives from $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ replacing route costs c_r with c'_r).

The procedure is initialized by setting $\boldsymbol{\lambda} = \mathbf{0}$ and $\boldsymbol{\sigma} = \mathbf{0}$.

At each iteration, for a given pair of penalty vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$, H^2 finds a solution $\overline{\boldsymbol{\xi}}$ of the *restricted problem* $\overline{LRP}(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ resulting from $LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ by replacing the set $\mathscr{R}$ with a limited subset $\overline{\mathscr{R}} \subset \mathscr{R}$. Note that $\overline{\boldsymbol{\xi}}$ might not be an optimal solution of the original problem $LRP'(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ and the dual vectors $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ associated to $\overline{\boldsymbol{\xi}}$ by means of expressions (43) might not be a feasible DRP' solution.

A route generation procedure, called GENROUTE, described in Sect. 6, is used to identify the route subset $\mathscr{N} \subset \mathscr{R} \backslash \overline{\mathscr{R}}$ whose DRP' constraints are violated by the current dual vectors $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$. If $\mathscr{N} \neq \emptyset$, then $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ is not a feasible DRP' solution and $\mathscr{N}$ is added to $\overline{\mathscr{R}}$. If $\mathscr{N} = \emptyset$, then $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ is a feasible DRP' solution and $LB_1 + z(\overline{LRP}(\boldsymbol{\lambda}, \boldsymbol{\sigma}))$ is a valid lower bound on the CVRP.

At each iteration, a subgradient vector is computed and used to update the penalty vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$ in order to maximize the value of lower bound LB_2.

Computing lower bound LB_2

H^2 uses the following parameters whose values are fixed *a priori*.

- Δ^{min}: the maximum size of the initial subset $\overline{\mathscr{R}}$ of routes.
- Δ^{a}: the maximum size of the subset $\mathscr{N}$ of routes violating DRP' constraints.
- Δ^{b}: the maximum size of $\overline{\mathscr{R}}$.
- $Maxt2$: the maximum number of iterations.

H^2 can be described as follows:

Step 1. Initialize $\lambda_i = 0, \forall i \in V'$, $\sigma_S = 0, \forall S \in \overline{\mathscr{S}}$ and $h = 1$.

Step 2. Generate an initial subset $\overline{\mathscr{R}} \subset \mathscr{R}$ containing the Δ^{min} routes of minimum reduced cost with respect to the DRP solution $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ obtained by H^1. Define $\overline{\mathscr{R}}_i = \overline{\mathscr{R}} \cap \mathscr{R}_i$, $i \in V$. If $|\overline{\mathscr{R}}_i| = 0$ for some vertex $i \in V$ then, add to $\overline{\mathscr{R}}$ the single customer route $R = (0, i, 0)$ of reduced cost $c'(R) = 2d_{0i} - u'_i - \sum_{S \in \overline{\mathscr{S}},\ S \ni i} v'_S$. Set $LB_2 = 0$, $\mathbf{u}^2 = \mathbf{0}$, $\mathbf{v}^2 = \mathbf{0}$, $\mathbf{w}^2 = \mathbf{0}$ and $\mathbf{g}^2 = \mathbf{0}$.

Step 3. Find a solution $\overline{\boldsymbol{\xi}}$ of the *restricted* problem $\overline{LRP}(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ resulting from $LRP(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ by replacing the set $\mathscr{R}$ with $\overline{\mathscr{R}}$. Let $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ be a not necessarily feasible DRP' solution associated to $\overline{\boldsymbol{\xi}}$ by means of expressions (43). If $z(\overline{LRP}(\boldsymbol{\lambda}, \boldsymbol{\sigma})) \leqslant LB_2$, go to Step 5.

Step 4. Using procedure GENROUTE generate the subset $\mathscr{N}$ of feasible routes having the largest negative reduced cost with respect to $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ and such that $|\mathscr{N}| \leqslant \Delta^a$. We have two cases:

(i) $\mathscr{N} = \emptyset$. $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ is a feasible DRP' solution. If $z(\overline{LRP}(\boldsymbol{\lambda}, \boldsymbol{\sigma})) > LB_2$, then update $LB_2 = z(\overline{LRP}(\boldsymbol{\lambda}, \boldsymbol{\sigma}))$ and set $\mathbf{u}^2 = \overline{\mathbf{u}}$, $\mathbf{v}^2 = \overline{\mathbf{v}}$.

(ii) $\mathscr{N} \neq \emptyset$. $(\overline{\mathbf{u}}, \overline{\mathbf{v}}, \overline{\mathbf{w}}, \overline{\mathbf{g}})$ is not a feasible DRP' solution. Update $\overline{\mathscr{R}} = \overline{\mathscr{R}} \cup \mathscr{N}$. If $|\overline{\mathscr{R}}| \geqslant \Delta^b$, remove from $\overline{\mathscr{R}}$ at most $|\overline{\mathscr{R}}| - \Delta^b$ routes having the largest positive reduced cost with respect to $(\mathbf{u}^2, \mathbf{v}^2, \mathbf{w}^2, \mathbf{g}^2)$.

Step 5. Set $h = h + 1$. If $h > Maxt2$, then stop; otherwise go to Step 6.

Step 6. Modify the penalty vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$ using expressions (56) and (57) as described in Sect. 5.2 for procedure H^1. Return to Step 3.

The initial route set $\overline{\mathscr{R}}$ at Step 2 and the subset $\mathscr{N}$, at each iteration of Step 4, are generated by procedure GENROUTE using different input parameters. At the end of H^2, $z'(DRP) = LB_1 + LB_2$ and the vectors $\mathbf{u}'$, $\mathbf{v}'$, $\mathbf{w}'$ and $\mathbf{g}'$, where $\mathbf{u}' = \mathbf{u}^1 + \mathbf{u}^2$, $\mathbf{v}' = \mathbf{v}^1 + \mathbf{v}^2$, $\mathbf{w}' = 0$ and $\mathbf{g}' = 0$, provide a lower bound on the CVRP and a feasible solution of DRP, respectively.

5.4 Bounding procedure H^3

Procedure H^3 computes lower bound LB_3 as the cost of the optimal solution of problem $\overline{RP}$ that derives from the LP-relaxation of RP by replacing the route costs c_r with the reduced costs c'_r computed with respect to the dual solution $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ achieved by H^1 and H^2.

We solve problem $\overline{RP}$ using a cut and column generation algorithm.

In order to prevent dual variables u_i, $i \in V$, from taking either very large or very small values we use a simple method, based on the Box Stabilization technique proposed by Marsten et al. [33]. The method consists of defining a box around the previous value of each dual variable u_i. At iteration, say t, we impose $u_i^{t-1} - L_i \leqslant u_i \leqslant u_i^{t-1} + L_i$, where u_i^{t-1} is the value of the dual variable u_i obtained at the previous iteration $t - 1$ and $L_i = q_i(z_{UB} - z'(DRP))/Q$, $i \in V$. In the first iteration we set $u_i^0 = 0$, $i \in V$.

The box imposed on each dual variable u_i can limit the value of lower bound LB_3 achieved by procedure H^3 but they allow to restrict the alternative optimal dual solutions of the master to those solutions having a more realistic distribution of the dual values. In practice, this avoids many of the iterations of standard cut and column generation algorithms, where dual variables taking extreme values can produce unrealistic routes of negative reduced cost to be added to the master.

Although this simple approach has show to perform quite well on the CVRP instances used in our computational experiments (see Sect. 7), several other stabilization methods have been proposed in the literature. For complete reviews and discussions on this topic, see Kim, Chang and Lee [25], du Merle et al. [34], Lübbecke and Desrosiers [30] and Rousseau et al. [41].

H^3 has several computational advantages with respect to standard cut and column generation methods. These latter algorithms are time-consuming as the LP-relaxation of the master problem is usually highly degenerate and degeneracy implies alternative optimal dual solutions. Consequently, the generation of new columns and their associated variables may not change the value of the objective function of the master problem, the master problem may become large, and the overall method may become slow computationally. H^3 starts from the near optimal dual solution of the LP-relaxation of RP produced by H^2. This allows us to generate an initial master problem containing the routes having a very small reduced cost that are likely to be in the optimal solution.

Computing lower bound LB_3

The initial master $\overline{RP}$ inherits from H^2 the final route subset $\overline{\mathscr{R}}$. We initialize $\overline{RP}$ by replacing the route set $\mathscr{R}$ with subset $\overline{\mathscr{R}}$, the set $\mathscr{S}$ with $\overline{\mathscr{S}}$ and we set $\mathscr{F} = \emptyset$ and $\mathscr{C} = \emptyset$.

At a given iteration (say t), let $\bar{\mathbf{y}}$ and $(\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{w}}, \bar{\mathbf{g}})$ be the optimal primal and dual solutions of problem $\overline{RP}$, respectively. We generate the largest subset $\mathscr{N} \subseteq \mathscr{R} \backslash \overline{\mathscr{R}}$ of routes of negative reduced cost with respect to $(\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{w}}, \bar{\mathbf{g}})$ such that $|\mathscr{N}| \leqslant \Delta^a$. If $\mathscr{N} = 0$, then procedure H^3 terminates, otherwise a new iteration is made. At iteration $t+1$, H^3 solves a new master problem $\overline{RP}$ by adding to $\overline{\mathscr{R}}$ the subset $\mathscr{N} \subseteq \mathscr{R} \backslash \overline{\mathscr{R}}$ of routes having negative reduced cost and the constraints violated by solution $\bar{\mathbf{y}}$.

Generation of route subset $\mathscr{N}$

The subset $\mathscr{N} \subseteq \mathscr{R} \backslash \overline{\mathscr{R}}$ of routes of negative reduced cost with respect to $(\bar{\mathbf{u}}, \bar{\mathbf{v}}, \bar{\mathbf{w}}, \bar{\mathbf{g}})$ is generated by means of procedure GENROUTE as described in Sect. 6.

Cut generation

The separation of violated inequalities is done after having added to $\overline{\mathscr{R}}$ the subset $\mathscr{N}$. Violated inequalities are separated as described in Sect. 3.1 and the sets $\overline{\mathscr{S}}$, $\mathscr{F}$ and $\mathscr{C}$ are updated accordingly.

At each iteration we limit each separation routine of the CVRPSEP package [31] to identify a maximum of 70 violated cuts. Moreover, we expand every new clique cut by adding routes of the current set $\overline{\mathscr{R}}$. This process can be time consuming as the size of the set $\overline{\mathscr{R}}$ is usually large. In order to reduce the computing time, in expanding a clique we use the $\hat{h}$ least reduced cost routes of $\overline{\mathscr{R}}$. In our computational results (see Sect. 7) we used $\hat{h} = 1,000$.

Termination criteria

Procedure H^3 terminates whenever $\mathscr{N} = \emptyset$. Let LB_3 and $(\mathbf{u}^3, \mathbf{v}^3, \mathbf{w}^3, \mathbf{g}^3)$ be the optimal solution cost and the optimal dual solution of $\overline{RP}$ achieved by H^3, respectively. $z'(DRP) = LB_1 + LB_2 + LB_3$ is the cost of the near optimal dual solution of $(\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{g}')$ of the LP-relaxation of RP obtained as $\mathbf{u}' = \mathbf{u}^1 + \mathbf{u}^2 + \mathbf{u}^3$, $\mathbf{v}' = \mathbf{v}^1 + \mathbf{v}^2 + \mathbf{v}^3$, $\mathbf{w}' = \mathbf{w}^3$ and $\mathbf{g}' = \mathbf{g}^3$.

6 A dynamic programming method for generating routes

In this section, we describe procedure GENROUTE for generating the initial set $\overline{\mathscr{R}}$ required at Step 1 of procedure H^2, the set $\mathscr{N}$ used by procedures H^2 and H^3 and the set $\mathscr{R}'$ required by the exact method described in Sect. 4. Procedure GENROUTE is an extension of the method proposed by Mingozzi et al. [36]. Given a DRP solution $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}}, \widehat{\mathbf{w}}, \widehat{\mathbf{g}})$ of cost $\widehat{z}(DRP)$, the subsets $\overline{\mathscr{S}}$, $\overline{\mathscr{F}}$ and $\overline{\mathscr{C}}$ of valid inequalities and two user-defined parameters γ and Δ, GENROUTE produces the largest subset $\mathscr{B}$ of the route set $\mathscr{R}$ satisfying the following conditions:

$$\left.\begin{array}{ll} \text{(a)} & \max\limits_{r\in\mathscr{B}}\{\widehat{c}_r\} \leqslant \min\limits_{r\in\mathscr{R}\setminus\mathscr{B}}\{\widehat{c}_r\} \\ \text{(b)} & |\mathscr{B}| \leqslant \Delta \\ \text{(c)} & \max\limits_{r\in\mathscr{B}}\{\widehat{c}_r\} \leqslant \gamma \end{array}\right\} \tag{58}$$

where $\widehat{c}_r$ is the reduced cost of route $r \in \mathscr{R}$ with respect to the DRP dual vectors $(\widehat{\mathbf{u}}, \widehat{\mathbf{v}}, \widehat{\mathbf{w}})$ but ignoring the dual vector $\widehat{\mathbf{g}}$. Parameters γ and Δ permit GENROUTE to generate the route subsets required by the following procedures described earlier in this paper:

- $\overline{\mathscr{R}}$ at Step 2 of H^2:
  define $\widehat{\mathbf{u}} = \mathbf{u}'$, $\widehat{\mathbf{v}} = \mathbf{v}'$, $\widehat{\mathbf{w}} = \mathbf{0}$ and set $\gamma = \infty$ and $\Delta = \Delta^{min}$.
- $\mathscr{N}$ at Step 3 of H^2:
  define $\widehat{\mathbf{u}} = \mathbf{u}' + \overline{\mathbf{u}}$, $\widehat{\mathbf{v}} = \mathbf{v}' + \overline{\mathbf{v}}$, $\widehat{\mathbf{w}} = \mathbf{0}$ and set $\gamma = 0$ and $\Delta = \Delta^a$.
- $\mathscr{N}$ required by H^3:
  define $\widehat{\mathbf{u}} = \mathbf{u}' + \overline{\mathbf{u}}$, $\widehat{\mathbf{v}} = \mathbf{v}' + \overline{\mathbf{v}}$, $\widehat{\mathbf{w}} = \overline{\mathbf{w}}$ and set $\gamma = 0$ and $\Delta = \Delta^a$.
- $\mathscr{R}'$ required by the exact method of Sect. 4:
  define $\widehat{\mathbf{u}} = \mathbf{u}'$, $\widehat{\mathbf{v}} = \mathbf{v}'$, $\widehat{\mathbf{w}} = \mathbf{w}'$, $\widehat{\mathbf{g}} = \mathbf{g}'$ and set $\gamma = z_{UB} - z'(DRP)$ and $\Delta = \infty$.

Procedure GENROUTE is based on the following observations.

(O1) Define the modified edge cost $\overline{d}_{ij}$ with respect to the dual vectors $\widehat{\mathbf{u}}$, $\widehat{\mathbf{v}}$ and $\widehat{\mathbf{w}}$ but ignoring $\widehat{\mathbf{g}}$ as follows:

$$\overline{d}_{ij} = d_{ij} - \frac{1}{2}\widehat{u}_i - \frac{1}{2}\widehat{u}_j - \frac{1}{2}\sum_{S\in\overline{\mathscr{S}}(i,j)} \widehat{v}_S - \sum_{t\in\overline{\mathscr{F}}} \alpha^t_{ij}\widehat{w}_t, \quad \forall\{i,j\}\in E. \tag{59}$$

Using the modified costs $\overline{d}_{ij}$, define the modified cost $\overline{c}_r$ of route $r \in \mathscr{R}$ as

$$\overline{c}_r = \sum_{\{i,j\}\in E(R_r)} \overline{d}_{ij}$$

It is easy to show that:

$$\overline{c}_r = c_r - \sum_{i\in V'} a_{ir}\widehat{u}_i - \sum_{S\in\overline{\mathscr{S}}} \left(\frac{1}{2}\sum_{\{i,j\}\in\delta(S)} \eta^r_{ij}\right)\widehat{v}_S - \sum_{t\in\overline{\mathscr{F}}} \alpha(R_r)\widehat{w}_t. \tag{60}$$

Consider the reduced cost $\widehat{c}_r$ of route r with respect to the four dual vectors $\widehat{\mathbf{u}}$, $\widehat{\mathbf{v}}$, $\widehat{\mathbf{w}}$ and $\widehat{\mathbf{g}}$ that is defined as:

$$\widehat{c}_r = c_r - \sum_{i \in V'} a_{ir}\widehat{u}_i - \sum_{S \in \overline{\mathscr{S}}} b_r(S)\widehat{v}_S - \sum_{t \in \overline{\mathscr{F}}} \alpha(R_r)\widehat{w}_t - \sum_{C \in \overline{\mathscr{C}}_r} \widehat{g}_C, \tag{61}$$

where $\overline{\mathscr{C}}_r = \{C \in \overline{\mathscr{C}} : C \ni r\}$.

Subtracting Eq. (60) from Eq. (61) we have:

$$\widehat{c}_r = \overline{c}_r + \sum_{S \in \overline{\mathscr{S}}} \left(\frac{1}{2} \sum_{\{i,j\} \in \delta(S)} \eta^r_{ij} - b_r(S) \right) \widehat{v}_S - \sum_{C \in \overline{\mathscr{C}}_r} \widehat{g}_C. \tag{62}$$

Since $b_r(S) \leqslant \frac{1}{2} \sum_{\{i,j\} \in \delta(S)} \eta^r_{ij}$ and $\widehat{\mathbf{g}} \leqslant \mathbf{0}$ from Eq. (62) we obtain:

$$\widehat{c}_r \geqslant \overline{c}_r, \quad \forall r \in \mathscr{R}. \tag{63}$$

(O2) Any route R can be decomposed, for every $i \in R$ into two paths P_i and $\overline{P}_i$ going from depot 0 to vertex i. These two paths are internally disjoint. (i.e., $V'(P_i) \cap V'(\overline{P}_i) = \{0, i\}$) and their total customer demand is less than or equal to Q, that is:

$$q(P_i) + q(\overline{P}_i) \leqslant Q + q_i, \quad \text{where } q(P) = \sum_{i \in V'(P)} q_i. \tag{64}$$

(O3) Any route R contains at least a vertex i (say) such that

$$\min\left\{q(P_i), q(\overline{P}_i)\right\} \geqslant \frac{q(R)}{2} \tag{65}$$

and

$$\max\left\{q(P_i), q(\overline{P}_i)\right\} \leqslant \frac{q(R)}{2} + q_i \tag{66}$$

In the following we call *median* of route R any vertex $i \in R$ satisfying inequalities (65) and (66).

(O4) Let $\mathscr{P}_i$ be the set of all simple paths ending at vertex $i \in V$ and such that $q(P) \leqslant Q/2 + q_i, \forall P \in \mathscr{P}_i, i \in V$. From observation (O3) it follows that any valid CVRP route can be obtained combining every pair of paths $P, \overline{P} \in \mathscr{P}_i$, $i \in V$, that are internally disjoint and such that:

$$\min\left\{q(P_i), q(\overline{P}_i)\right\} \geqslant \frac{q(P) + q(\overline{P}) - q_i}{2}$$

$$\max\left\{q(P_i), q(\overline{P}_i)\right\} \leqslant \frac{q(P) + q(\overline{P}) - q_i}{2} + q_i.$$

(O5) Using the modified edge cost $\overline{d}_{ij}$ defined by expressions (59), let $\overline{c}(P)$ be the cost of path P, and let $LB(P)$ be a lower bound to the cost $\overline{c}(R)$ of any route R obtained by combining P with any path $\overline{P}$ such that:
  (i) P and $\overline{P}$ are internally disjoint;
  (ii) P and $\overline{P}$ have the same terminal vertex i;
  (iii) $q(P) + q(\overline{P}) \leqslant Q + q_i$.
  Since $\overline{c}(R) \leqslant \widehat{c}(R)$ for any route R (see observation (O1)), than any path P such that $LB(P) > \gamma$ cannot produce any route of the set $\mathscr{B}$ satisfying conditions (58). A method for computing lower bound $LB(P)$ is described in Sect. 6.1.

From these observations, we derive the following two phases method for generating the route set $\mathscr{B}$.

Phase 1: *generation of the path set* $\mathscr{P}$

Using the modified edge costs $\overline{d}_{ij}$ defined by expressions (59) generate the largest set $\mathscr{P}$ of simple paths such that:

(a) $\max\limits_{P\in\mathscr{P}}\{LB(P)\} \leqslant \min\limits_{P\notin\mathscr{P}}\{LB(P), \gamma\}$;
(b) $|\mathscr{P}| \leqslant \Delta$;
(c) $q(P) \leqslant \dfrac{Q}{2} + q_{e(P)}, \forall P \in \mathscr{P}$, where $e(P)$ is the terminal vertex of P;
(d) a path P' cannot belong to $\mathscr{P}$ if $\mathscr{P}$ contains another path P such that $e(P') = e(P')$, $V'(P') = V'(P)$ and $c(P') \geqslant c(P)$.

In Sect. 6.1 we describe a procedure, called GENPATH, for generating the set $\mathscr{P}$.

Phase 2: *generation of the route set* $\mathscr{B}$

Define $\omega = \max_{P\in\mathscr{P}}\{LB(P)\}$. It is easy to observe that $\omega \leqslant \gamma$. Moreover, it is quite easy to show that any route R of reduced cost $\widehat{c}(R) \leqslant \omega$ can be obtained combining two paths $P, \overline{P} \in \mathscr{P}_i$ for some $i \in R\backslash\{0\}$.

We say that route R' is *dominated* by route R if $R = R'$ and $c(R') \geqslant c(R)$.

A simple, but computationally expensive, method for generating a subset $\mathscr{B}$ of routes satisfying conditions (58) is based on the following two steps procedure.

(A) Generate the set $\widehat{\mathscr{B}}$ of *undominated* routes by combining every pair of paths $P, \overline{P} \in \mathscr{P}$ having the same terminal vertex i and such that $V'(P) \cap V'(\overline{P}) = \{0, i\}$, $q(P) + q(\overline{P}) \leqslant Q + q_i$ and $\overline{c}(P) + \overline{c}(\overline{P}) \leqslant \omega$.
(B) Extract from $\widehat{\mathscr{B}}$ the largest subset of routes $\mathscr{B}$ satisfying conditions (58). This requires the computation of the reduced cost $\widehat{c}_r, \forall r \in \widehat{\mathscr{B}}$ and the removal from $\widehat{\mathscr{B}}$ of any route r such that $\widehat{c}_r > \gamma$.
  This method is computationally expensive since it involves the generation of all feasible routes that can be obtained combining every pair of paths $P, \overline{P} \in \mathscr{P}$ with $e(P) = e(\overline{P})$. A more efficient method for generating $\mathscr{B}$ is procedure GENROUTE described in Sect. 6.2.

6.1 Procedure GENPATH

This procedure generates the set $\mathscr{P}$ of simple paths satisfying conditions (a),(b),(c) and (d) in the previous section.

Associate with the undirected graph G, the directed graph $G' = (V', A)$ containing two arcs (i, j) and (j, i), $\forall\{i, j\} \in E$, of cost $\overline{d}_{ij}$ and $\overline{d}_{ji} = \overline{d}_{ij}$, respectively.

This procedure is analogous to Dijkstra's algorithm on an expanded state-space graph that is dynamically generated. Each vertex of this state-space corresponds to a feasible path. Let T be the set of "temporary" paths.

At each iteration, GENPATH extracts from T the path P^* of minimum lower bound (i.e., $LB(P^*) = \min\{LB(P) : P \in T\}$) and inserts P^* in $\mathscr{P}$. The selected path P^* is then expanded. Each resulting path P' that is not dominated by some other path in $T \cup \mathscr{P}$ is inserted in T and any path $P \in T$ that is dominated by P' is removed from T. The procedure terminates when either $T = \emptyset$ or $|\mathscr{P}| = \Delta$.

In order to avoid *out-of-memory* errors, we impose that the size of the set T does not exceed an a priori defined memory limit $NSTATB$. If $|T|$ becomes greater than $NSTATB$, then GENPATH and the entire algorithm terminate prematurely.

Below we give a step-by-step description of procedure GENPATH, followed by the description of the method used for computing lower bound $LB(P)$.

Description of procedure GENPATH

Step 1. Let $P = (0)$ be the path containing only the depot. Set $T = \{P\}$, $\overline{c}(P) = 0$, $e(P) = 0$. Compute $LB(P)$ and initialize $\mathscr{P} = \emptyset$.

Step 2. If $T = \emptyset$, then Stop.

Step 3. Let $P^* \in T$ be such that $LB(P^*) = \min\{LB(P) : P \in T\}$. Update $T = T\backslash\{P^*\}$ and $\mathscr{P} = \mathscr{P} \cup \{P^*\}$. If $|\mathscr{P}| = \Delta$ then Stop. If $q(P^*) > Q/2$ return to Step 3.

Step 4. For every vertex $j \in V\backslash V'(P^*)$ repeat the following Step 5.

Step 5. Let P' be the path obtained by appending arc $\{e(P^*), j\}$ at the end of P^*. Compute the cost $\overline{c}(P') = \overline{c}(P^*) + \overline{d}_{e(P^*)j}$ and the lower bound $LB(P')$ as described in Sect. 6.1. The expanded path P' can be rejected because of one of the following tests:

A. *Lower bound test.* $LB(P') \geqslant \gamma$.

B. *Dominance test.* There exists an other path $P \in T \cup \mathscr{P}$ such that $e(P) = e(P')$, $V'(P) = V'(P')$ and $c(P) \leqslant c(P')$ where $c(P)$ is the cost of path P using costs d_{ij}.

If P' is not rejected by tests A or B, then remove from T any path P that is dominated by P' (i.e., $e(P) = e(P')$, $V'(P) = V'(P')$ and $c(P) \geqslant c(P')$). If $|T| < NSTATB$ then insert P' in T, otherwise Stop.

Step 6. Return to Step 2.

In our computational results (see Sect. 7) only three instances out of 75 could not be solved using $NSTATB = 10 \times 10^6$.

Computing lower bound $LB(P)$

Lower bound $LB(P)$ can be computed using the q-path functions $f(q,i)$, $\pi(q,i)$ and $g(q,i)$ described in Christofides et al. [15]. More precisely, let $f(q,i)$ be the cost of the least cost q-path ending at vertex i and let $\pi(q,i)$ be the vertex just prior to i. Let $g(q,i)$ be the cost of the least cost q-path ending at vertex i with the constraint that the vertex $\gamma(q,i)$ preceding i is not equal to $\pi(q,i)$. These functions must be computed using the arc costs $\overline{d}_{ij}$ before starting GENPATH. As any route containing P must contain another path $\overline{P}$ terminating in $e(P)$ of load q' such that $q_{e(P)} \leqslant q' \leqslant Q - q(\overline{P}) + q_{e(P)}$, we have:

$$LB(P) = \overline{c}(P) + \min_{q_{e(P)} \leqslant q' \leqslant Q + q(P) + q_{e(P)}} \left\{ \begin{array}{ll} f(q', e(P)), \text{if} \pi(q', e(P)) \notin V'(P)) \\ g(q', e(P)), & \text{otherwise} \end{array} \right\}. \tag{67}$$

The computation of $LB(P)$ by means of expression (67) can be time consuming. A better method for computing $LB(P)$ consists of avoiding the minimization required in the right-hand-side of expression (67) as follows.

Let $F(q,i)$ be the least cost of the least cost q-path of load less than or equal $Q - q + q_i$ ending at vertex i and let $\chi(q,i)$ be the vertex prior to i in such a q-path. Let $G(q,i)$ be the cost of the least cost q-path of load less than or equal to $Q - q + q_i$ ending at vertex i and such that the vertex preceding i is not equal to $\chi(q,i)$. Functions $F(q,i)$, $\chi(q,i)$ and $G(q,i)$ can be computed using functions $f(q,i)$,$\pi(q,i)$ and $g(q,i)$ before starting GENPATH, as follows:

$$F(q,i) = \min_{q_i \leqslant q' \leqslant Q - q + q_i} \{f(q',i)\}, \quad \forall i \in V, \forall q : q_i \leqslant q \leqslant Q \tag{68}$$

and let $\chi(q,i) = \pi(q^*,i)$ where q^* is the value of q' giving the minimum in expression (68).

$$G(q,i) = \min_{q_i \leqslant q' \leqslant Q - q + q_i} \left\{ \begin{array}{ll} f(q',i), & \text{if } \pi(q',i) \notin \chi(q',i) \\ g(q',i), & \text{otherwise} \end{array} \right\}, \\ \forall i \in V, \forall q : q_i \leqslant q \leqslant Q. \tag{69}$$

Using functions $F(q,i)$, $\chi(q,i)$ and $G(q,i)$, we have

$$LB(P) = c'(P) + \left[\begin{array}{ll} F(q(P), e(P)), & \text{if } \chi(q(P), e(P)) \notin V'(P)) \\ G(q(P), e(P)), & \text{otherwise} \end{array} \right].$$

6.2 Procedure GENROUTE

This procedure starts with $\mathscr{B} = \emptyset$ and, iteratively, adds to set $\mathscr{B}$ the route having minimum reduced cost among those routes not yet in $\mathscr{B}$. GENROUTE terminates when either $|\mathscr{B}| = \Delta$ or when the cost $\overline{c}(R)$ of the route R candidate to enter $\mathscr{B}$ is greater than ω defined in Sect. 6.

For the sake of simplicity, in the description of procedure GENROUTE, we denote by P_i a path ending at vertex i. Procedure GENROUTE requires that each set $\mathscr{P}_i$ is ordered for non-decreasing values of the path costs (i.e., $\mathscr{P}_i = (P_i^1, P_i^2, \ldots, P_i^{h_i})$ where $\overline{c}(P_i^1) \leqslant \overline{c}(P_i^2) \leqslant \cdots \leqslant \overline{c}(P_i^{h_i})$).

Procedure GENROUTE dynamically generates the set T that contains a subset of all possible pairs of paths $(P, \overline{P})$, with $P, \overline{P} \in \mathscr{P}$ and $e(P) = e(\overline{P})$.

A cost equal to $\overline{c}(P) + \overline{c}(\overline{P})$ is associated with each pair of paths $(P, \overline{P}) \in T$. At each iteration, GENROUTE selects from set T the pair of paths of minimum cost and, if these two paths correspond to a feasible route, then it adds such route to the emerging set $\mathscr{B}$. The selected pair of paths is then used to expand the set T.

A step by step description of procedure GENROUTE is the following.

Description of procedure GENROUTE

Step 1. Initialize $\mathscr{B} = \emptyset$. Define $T = \{(P_i^1, P_i^2) : i \in V$ such that $|\mathscr{P}_i| > 1\}$.

Step 2. If $T = \emptyset$, then Stop.

Step 3. Let $(P_i^h, P_i^s) \in T$ be such that $\overline{c}(P_i^h)+\overline{c}(P_i^s) = \min\{\overline{c}(P)+\overline{c}(\overline{P}) : (P, \overline{P}) \in T\}$. Remove (P_i^h, P_i^s) from T. The two paths P_i^h and P_i^s produces a route R that can be added to set $\mathscr{B}$ if the following conditions are satisfied:
- (i) $V'(P_i^h) \cap V'(P_i^s) = \{0, i\}$;
- (ii) $\widehat{c}(R) \leqslant \gamma$;
- (iii) vertex i is a median of route R;
- (iv) R is not dominated by any other route $R' \in \mathscr{B}$.

If $|\mathscr{B}| = \Delta$, then Stop.

Step 4. Add the following pairs of paths to T.
- (P_i^h, P_i^{s+1}), if $\overline{c}(P_i^h) + \overline{c}(P_i^{s+1}) \leqslant \omega$, $h < s$ and $s < |\mathscr{P}_i|$.
- (P_i^{h+1}, P_i^s), if $\overline{c}(P_i^{h+1}) + \overline{c}(P_i^s) \leqslant \omega$, and $s - h = 2$.

Return to Step 2.

Algorithm GENROUTE can be easily adapted to deal with other real-world route constraints (time windows, capacity constraints, etc.) simply by rejecting any infeasible route at Step 3.

7 Computational results

This section presents computational results of the exact method and of the bounding procedure HDRP described in Sects. 4 and 5, respectively.

The algorithms described in this paper were coded in Fortran 77, compiled with Compaq Digital Fortran 6.6 compiler and linked with the C source codes of the packages CVRPSEP [31] and CLIQUER [39]. CPLEX 9.0 [18] was used as the LP solver in procedure H^3 and as the integer programming solver in the exact method. The experiments were performed on a Pentium 4 personal computer at 2.6 GHz equipped with 3 Gb of RAM and running under Windows XP.

We have considered five classes of test instances taken from the literature called A, B, E, M and P, respectively. Classes A, B and P were proposed by Augerat [1].

Instance class M was proposed by Christofides et al. [14] while class E was produced by Christofides and Eilon [13].

The data of all instances including the best upper bounds known and the solutions of the instances solved to optimality can be found at the Internet address `http://branchandcut.org/VRP/data`. In the following we use the same naming scheme for the instance data adopted by Augerat et al. [2]. For all the instances considered, the edge cost d_{ij} is an integer value computed as $d_{ij} = \lfloor e_{ij} + 0.5 \rfloor$, where e_{ij} is the Euclidean distance between vertices i and j.

We made several preliminary experiments to identify good parameter setting for our method. As a result of such experiments we decided to use the following setting of the parameters:

- in procedure H^1: $Maxt1 = 500$, $\epsilon = 2.0$;
- in procedure H^2: $Maxt2 = 25$, $\epsilon = 1.0$, $\Delta^{min} = 5000$, $\Delta^a = 200$, $\Delta^b = 10000$;
- in procedure H^3: $\Delta^a = 4000$;
- in the exact method described in Sect. 4, we disabled the separation of all the cuts embedded into CPLEX, since their use increases the overall computing time.

In the following we denote by BCP the branch-and-cut-and-price algorithm of Fukasawa et al. [21] and by ESP the exact set partitioning method described in Sect. 4.

The tables of this section report the following columns:

- z^* : cost of the optimal CVRP solution;
- z_{UB} : upper bound value used by both BCP and ESP;
- LB_B : lower bound obtained at the root node of BCP;
- $\%LB_B$: percentage ratio of lower bound LB_B computed as $100.0\ LB_B/z^*$;
- t_{LB_B} : time in seconds of a Pentium 4 running at 2.4 Ghz for computing lower bound LB_B;
- s : size of the cycles eliminated by the column generation procedure of BCP. "–" indicates that column generation was not used by BCP and, in this case, BCP becomes the branch-and-cut algorithm of Lysgaard et al. [32] ;
- t_B : total computing time in seconds of BCP on a Pentium 4 running at 2.4 Ghz;
- $\%LB'_h, h = 1, 2, 3$: percentage ratio of the lower bound $z'(DRP)$ achieved by the additive bounding procedure HDRP at the end of iteration t, that is $LB'_1 = LB_1$, $LB'_2 = LB_1 + LB_2$ and $LB'_3 = LB_1 + LB_2 + LB_3$ (see Sects. 5.2, 5.3 and 5.4 computed as $100.0\ LB'_t/z^*, t = 1, 2, 3$.
- t_{HDRP} : time in seconds spent by HDRP for computing the final lower bound LB'_3; t_{HDRP} includes also the time spent for computing the initial set of capacity constraints $\overline{\mathscr{S}}$;
- $|\mathscr{R}'|$: final number of routes generated by our exact method (see Sect. 4);
- t_{CPX} : total computing time in seconds spent by CPLEX [18] to solve the final RP problem; t_{CPX} includes also the time spent by procedure GENROUTE for generating the final route set $\mathscr{R}'$;

t_{ESP} : total computing time in seconds of the exact method. This includes the computing times for generating the set of capacity constraints $\overline{\mathscr{S}}$ used by H^1 and H^2, for computing the lower bounds, for generating the route set $\mathscr{R}'$ and for solving the final RP problem using CPLEX.

To evaluate the contributions given by the different types of valid inequalities to the final lower bound LB'_3, we have performed a number of experiments using a subset of 15 hard CVRP instances. Table 1 reports for each instance considered, the lower bound obtained by Fukasawa et al., and the lower bounds obtained by considering in procedure H^3 (i) capacity constraints only (under column heading "H^3 (CC)"), (ii) capacity and clique inequalities (under column heading "H^3 (CC+CI)") and (iii) capacity, clique, framed capacity, strengthened comb and hypotour inequalities (under column heading "H^3 (All cuts)"). In particular, for the computation of the lower bounds in procedure H^3 using capacity and clique inequalities, first we separated the capacity constraints and clique inequalities were separated only when no violated capacity inequalities were found. For the computation of the lower bounds using all inequalities, we proceeded to the separation of all the remaining classes of inequalities whenever no violated capacity constraints and clique inequalities were identified. Following the computational results reported in Table 1, we decided to use in procedure H^3 capacity and clique inequalities only, as the improvement to the lower bound given by the other valid inequalities is not worth the extra computing time required.

Tables 2, 3, 4 and 5 report the lower bounds obtained by procedures H^1, H^2 and H^3 and the results of the exact method for the classes of test instances A, B, E, M and P.

The last line of each table reports:

- the percentage average ratios and the average running times in seconds of the lower bounds computed over all instances;
- the average running times in seconds computed over all instances solved to optimality by both methods.

A comparison between lower bound LB_B and lower bound LB'_3 shows that on average LB'_3 is superior to LB_B in all instance classes. In particular, LB'_3 is superior to LB_B in all instances except instance P-n101-k4 for which procedure H^3 prematurely terminated since the maximum size of the set T was reached in procedure GENPATH.

The stabilization technique described in Sect. 5.4 was very effective in reducing the computing time of procedure H^3 only for instances of classes M and P involving more than 100 and 70 customers, respectively.

Since our Pentium 4 2.6 GHz is about 10% faster than the Pentium 4 2.4 GHz used by Fukasawa et al., Tables 2, 4 and 5 indicate that our bounding procedure is on average faster than the one of Fukasawa et al. on instance classes A, E, M and P but not for class B instances, where 14 instances out of 20 were solved by BCP without using column generation.

The results of Tables 2, 3, 4 and 5 concerning the two exact methods can be analyzed as follows. The new exact algorithm ESP solved to optimality all instances of classes A and B but it cannot optimally solve instance E-n101-k8 of class E, and instances P-n76-k5 and P-n101-k4 of class P. These instances, that are loosely constrained,

Table 1 Contribution to the lower bound of different types of valid inequalities

			Fukasawa et al. BCP			H^3 (CC)			H^3 (CI)			H^3 (CC+CI)			H^3 (All cuts)		
Instance	z_{UB}	z^*	LB_B	$\%LB_B$	t_{LB_B}	LB'_3	$\%LB'_3$	t_{HDRP}	LB'_3	$\%LB'_3$	t_{HDRP}	LB'_3	$\%LB'_3$	t_{HDRP}	LB'_3	$\%LB'_3$	t_{HDRP}
A-n63-k10	1,315	1,314	1,299.1	98.9	136.0	1,303.8	99.2	12.7	1,298.6	98.8	19.6	1,309.5	99.7	24.8	1,309.6	99.7	31.3
A-n64-k9	1,402	1,401	1,385.3	98.9	265.0	1,390.5	99.3	16.8	1,393.7	99.5	24.9	1,394.2	99.5	32.1	1,396.5	99.7	172.9
A-n69-k9	1,159	1,159	1,141.4	98.5	289.0	1,145.1	98.8	13.3	1,152.6	99.4	43.2	1,153.9	99.6	72.2	1,156.6	99.8	306.7
A-n80-k10	1,763	1,763	1,754.0	99.5	1,120.0	1,756.5	99.6	74.7	1,753.5	99.5	117.6	1,756.5	99.6	117.1	1,762.5	100.0	370.2
B-n50-k8	1,312	1,312	1,295.0	98.7	97.0	1,303.5	99.3	452.7	1,306.7	99.6	674.4	1,307.6	99.7	639.7	1,307.6	99.7	665.8
B-n68-k9	1,275	1,272	1,263.0	99.3	260.0	1,263.8	99.4	196.0	1,265.9	99.5	241.7	1,266.1	99.5	254.2	1,266.1	99.5	265.4
E-n51-k5	521	521	518.2	99.5	51.0	518.1	99.4	10.0	521.0	100.0	11.6	521.0	100.0	12.6	521.0	100.0	12.6
E-n76-k7	682	682	670.0	98.2	264.0	670.0	98.2	20.8	675.3	99.0	94.8	675.4	99.0	146.4	676.1	99.1	396.1
E-n76-k8	735	735	726.5	98.8	277.0	725.7	98.7	19.0	727.2	98.9	79.0	729.8	99.3	103.5	731.6	99.5	308.2
E-n76-k10	830	830	817.4	98.5	354.0	817.2	98.5	14.3	825.4	99.4	65.6	825.8	99.5	59.6	826.7	99.6	431.5
E-n76-k14	1,021	1,021	1,006.5	98.6	224.0	1,007.9	98.7	11.2	1,016.3	99.5	17.6	1,016.5	99.6	16.8	1,016.5	99.6	31.6
E-n101-k8	815	815	805.2	98.8	1,068.0	804.4	98.7	56.1	806.4	98.9	120.5	807.2	99.0	249.9	808.3	99.2	1,431.9
E-n101-k14	1,071	1,067	1,053.8	98.8	658.0	1,058.8	99.2	33.0	1,061.7	99.5	96.2	1,064.3	99.7	153.5	1,065.2	99.8	712.5
P-n50-k8	649	631	616.3	97.7	102.0	617.9	97.9	10.9	624.4	98.9	13.8	624.7	99.0	11.8	624.7	99.0	14.8
P-n70-k10	834	827	814.5	98.5	292.0	814.3	98.5	16.2	821.2	99.3	26.3	821.7	99.4	35.1	821.8	99.4	63.8
Averages				98.7	363.8		98.9	63.8		99.3	109.8		99.5	128.6		99.6	347.7

Table 2 Computational results for the class A instances

Instance	z_{UB}	z^*	Fukasawa et al. BCP LB_B	$\%LB_B$	t_{LB_B}	s	t_B	Bounding procedure **HDRP** LB'_1	$\%LB'_1$	LB'_2	$\%LB'_2$	LB'_3	$\%LB'_3$	t_{HDRP}	Exact Method ESP $\lvert\mathscr{R}'\rvert$	t_{CPX}	t_{ESP}
A-n37-k5	669	669	664.8	99.4	16.0	–	19.0	643.1	96.1	659.6	98.6	669.0	100.0	13.0	0	0.0	13.0
A-n37-k6	949	949	932.6	98.3	30.0	3	379.0	910.1	95.9	931.9	98.2	945.0	99.6	9.9	3,176	2.4	12.3
A-n38-k5	730	730	716.5	98.2	12.0	–	26.0	674.9	92.5	702.0	96.2	726.7	99.6	15.0	5,494	3.0	18.1
A-n39-k5	822	822	816.6	99.3	107.0	3	167.0	801.2	97.5	807.8	98.3	822.0	100.0	19.8	0	0.0	19.8
A-n39-k6	831	831	822.8	99.0	39.0	3	98.0	792.2	95.3	812.1	97.7	829.2	99.8	10.2	4,584	2.4	12.6
A-n44-k6	937	937	934.8	99.8	52.0	2	90.0	917.8	97.9	931.6	99.4	937.0	100.0	102.4	0	0.0	102.4
A-n45-k6	944	944	938.1	99.4	52.0	3	170.0	895.2	94.8	932.0	98.7	944.0	100.0	77.8	0	0.0	77.8
A-n45-k7	1,146	1,146	1,139.3	99.4	88.0	3	331.0	1,120.7	97.8	1,128.5	98.5	1,145.4	100.0	35.3	0	0.0	35.3
A-n46-k7	914	914	914.0	100.0	63.0	2	92.0	895.5	98.0	907.9	99.3	914.0	100.0	11.8	0	0.0	11.8
A-n48-k7	1,073	1,073	1,069.1	99.6	72.0	3	166.0	1,039.8	96.9	1,053.9	98.2	1,073.0	100.0	20.3	0	0.0	20.3
A-n53-k7	1,010	1,010	1,003.9	99.4	138.0	3	611.0	979.5	97.0	995.5	98.6	1,008.2	99.8	22.9	10,145	5.1	28.0
A-n54-k7	1,167	1,167	1,153.9	98.9	125.0	3	1,409.0	1,117.8	95.8	1,144.1	98.0	1,161.4	99.5	43.1	12,721	43.5	86.6
A-n55-k9	1,073	1,073	1,067.4	99.5	32.0	3	84.0	1,026.7	95.7	1,060.6	98.8	1,071.5	99.9	17.2	4,478	1.8	19.1
A-n60-k9	1,354	1,354	1,344.4	99.3	161.0	3	3,080.0	1,313.5	97.0	1,333.5	98.5	1,351.2	99.8	90.9	7,704	20.2	111.1
A-n61-k9	1,034	1,034	1,022.5	98.9	108.0	3	1,883.0	996.6	96.4	1,014.1	98.1	1,029.1	99.5	21.2	6,928	13.8	34.9
A-n62-k8	1,290	1,288	1,280.4	99.4	722.0	3	3,102.0	1,235.1	95.9	1,267.6	98.4	1,286.5	99.9	70.8	56,482	1,271.5	1,342.3
A-n63-k9	1,616	1,616	1,607.0	99.4	238.0	3	1,046.0	1,572.8	97.3	1,595.6	98.7	1,615.2	99.9	34.4	7,359	9.7	44.1
A-n63-k10	1,315	1,314	1,299.1	98.9	136.0	3	4,988.0	1,268.6	96.5	1,286.8	97.9	1,309.5	99.7	24.8	9,325	105.1	129.9
A-n64-k9	1,402	1,401	1,385.3	98.9	265.0	3	11,254.0	1,360.2	97.1	1,381.3	98.6	1,394.2	99.5	32.1	28,837	88.9	120.9
A-n65-k9	1,174	1,174	1,166.5	99.4	154.0	3	516.0	1,135.4	96.7	1,151.2	98.1	1,174.0	100.0	38.5	0	0.0	38.5
A-n69-k9	1,159	1,159	1,141.4	98.5	289.0	3	7,171.0	1,120.5	96.7	1,139.5	98.3	1,153.9	99.6	72.2	14,613	52.3	124.5
A-n80-k10	1,763	1,763	1,754.0	99.5	1,120.0	3	6,464.0	1,723.0	97.7	1,742.9	98.9	1,756.5	99.6	117.1	31,104	77.1	194.2
Averages				99.2	182.7		1,961.2		96.5		98.4		99.8	40.9			118.1

Table 3 Computational results for the class B instances

Instance	z_{UB}	z^*	Fukasawa et al. BCP LB_B	$\%LB_B$	t_{LB_B}	s	t_B	Bounding procedure **HDRP** LB'_1	$\%LB'_1$	LB'_2	$\%LB'_2$	LB'_3	$\%LB'_3$	t_{HDRP}	Exact Method ESP $\lvert\mathscr{R}'\rvert$	t_{CPX}	t_{ESP}
B-n38-k6	805	805	800.2	99.4	10.0	–	14.0	801.6	99.6	803.9	99.9	805.0	100.0	16.9	101	0.0	16.9
B-n39-k5	549	549	549.0	100.0	3.0	–	3.0	543.2	98.9	549.0	100.0	549.0	100.0	7.6	0	0.0	7.6
B-n41-k6	829	829	826.4	99.7	13.0	–	18.0	823.4	99.3	828.6	100.0	828.8	100.0	15.8	34	0.0	15.8
B-n43-k6	742	742	733.7	98.9	13.0	–	29.0	735.6	99.1	736.9	99.3	738.2	99.5	34.8	13,262	2.2	37.0
B-n44-k7	909	909	909.0	100.0	9.0	–	9.0	909.0	100.0	909.0	100.0	909.0	100.0	24.4	15	0.0	24.4
B-n45-k5	751	751	747.5	99.5	10.0	–	16.0	746.2	99.4	750.9	100.0	751.0	100.0	16.2	14	0.0	16.2
B-n45-k6	678	678	677.5	99.9	224.0	3	279.0	672.6	99.2	676.7	99.8	677.9	100.0	61.0	195	0.0	61.0
B-n50-k7	741	741	741.0	100.0	5.0	–	6.0	739.1	99.7	741.0	100.0	741.0	100.0	24.7	20	0.0	24.7
B-n50-k8	1,312	1,312	1,295.0	98.7	97.0	3	2,845.0	1,288.8	98.2	1,302.8	99.3	1,307.6	99.7	639.7	50,632	22.6	662.3
B-n51-k7	1,032	1,032	1,025.2	99.3	16.0	–	46.0	1,010.0	97.9	1,027.0	99.5	1,027.4	99.6	51.3	8,809	0.9	52.2
B-n52-k7	747	747	745.8	99.8	7.0	–	9.0	737.4	98.7	745.4	99.8	747.0	100.0	29.1	341	0.0	29.1
B-n56-k7	707	707	704.0	99.6	15.0	–	22.0	703.9	99.6	705.0	99.7	706.1	99.9	73.7	401	0.0	73.7
B-n57-k7	1,153	1,153	1,149.2	99.7	76.0	–	168.0	1,123.3	97.4	1,151.0	99.8	1,153.0	100.0	184.0	1,326	0.0	184.0
B-n57-k9	1,598	1,598	1,596.0	99.9	61.0	3	193.0	1,592.6	99.7	1,595.0	99.8	1,597.8	100.0	60.9	762	0.0	60.9
B-n63-k10	1,496	1,496	1,479.4	98.9	231.0	–	682.0	1,480.6	99.0	1,485.3	99.3	1,491.7	99.7	93.7	12,842	0.5	94.1
B-n64-k9	861	861	859.3	99.8	70.0	–	86.0	859.2	99.8	860.3	99.9	860.3	99.9	70.6	391	0.0	70.6
B-n66-k9	1,316	1,316	1,307.5	99.4	145.0	3	1,778.0	1,290.0	98.0	1,307.4	99.3	1,309.9	99.5	215.9	27,958	11.9	227.8
B-n67-k10	1,032	1,032	1,024.4	99.3	218.0	–	568.0	1,025.6	99.4	1,026.7	99.5	1,029.4	99.7	282.5	13,713	4.7	287.1
B-n68-k9	1,275	1,272	1,263.0	99.3	260.0	3	87,436.0	1,258.2	98.9	1,263.2	99.3	1,266.1	99.5	254.2	440,289	5,913.8	6,168.0
B-n78-k10	1,221	1,221	1,215.2	99.5	193.0	3	1,053.0	1,207.5	98.9	1,211.8	99.2	1,218.8	99.8	227.6	14,058	2.4	229.9
Averages				99.5	83.8		4,763.0		99.0		99.7		99.8	119.2			417.2

Table 4 Computational results for the classes E and M instances

			Fukasawa et al. BCP					Bounding procedure **HDRP**							Exact Method ESP		
Instance	z_{UB}	z^*	LB_B	$\%LB_B$	t_{LB_B}	s	t_B	LB'_1	$\%LB'_1$	LB'_2	$\%LB'_2$	LB'_3	$\%LB'_3$	t_{HDRP}	$\lvert\mathscr{R}'\rvert$	t_{CPX}	t_{ESP}
E-n51-k5	521	521	518.2	99.5	51.0	–	65.0	514.0	98.7	517.1	99.3	521.0	100.0	12.6	0	0.0	12.6
E-n76-k7	682	682	670.0	98.2	264.0	2	46,520.0	667.5	97.9	669.5	98.2	675.4	99.0	146.4	567,413	3,224.3	3,370.6
E-n76-k8	735	735	726.5	98.8	277.0	2	22,891.0	720.5	98.0	721.7	98.2	729.8	99.3	103.5	89,222	769.7	873.2
E-n76-k10	830	830	817.4	98.5	354.0	3	80,722.0	811.6	97.8	813.8	98.0	825.8	99.5	59.6	28,375	114.8	174.4
E-n76-k14	1,021	1,021	1,006.5	98.6	224.0	3	48,637.0	999.5	97.9	1,002.8	98.2	1,016.5	99.6	16.8	9,084	28.1	44.9
E-n101-k8	815	815	805.2	98.8	1,068.0	3	801,963.0	796.9	97.8	802.8	98.5	807.2	99.0	249.9	>2,000,000	–	[a]
E-n101-k14	1,071	1,067	1,053.8	98.8	658.0	3	116,284.0	1,045.1	97.9	1,050.5	98.5	1,064.3	99.7	153.5	76,756	1,076.4	1,230.0
M-n101-k10	820	820	820.0	100.0	119.0	–	119.0	809.8	98.8	819.5	99.9	820.0	100.0	47.0	0	0.0	47.0
M-n121-k7	1,034	1,034	1,031.1	99.7	5,594.0	3	25,678.0	1,028.2	99.4	1,032.2	99.8	1,032.2	99.8	943.7	66,819	1,504.6	2,448.3
M-n151-k12	1,015	–	999.1	98.4	945.0	3	–	993.5	97.9	993.7	97.9	1,003.0	98.8	356.5	–	–	–
M-n200-k16	–	–	1,252.4	–	3,168.0	3	–	1,240.3	–	1,251.2	–	1,256.4	–	276.9	–	–	–
M-n200-k17	1,275	–	1,254.2	98.4	2,310.0	3	–	1,243.2	97.5	1,253.2	98.3	1,256.7	98.6	315.0	–	–	–
Averages				98.9	1,252.7		42,614.5		98.1		98.6		99.4	223.4			1,025.1

[a] CPLEX out of memory

Table 5 Computational results for the class P instances

Instance	z_{UB}	z^*	Fukasawa et al. BCP LB_B	$\%LB_B$	t_{LB_B}	s	t_B	Bounding procedure **HDRP** LB'_1	$\%LB'_1$	LB'_2	$\%LB'_2$	LB'_3	$\%LB'_3$	t_{HDRP}	Exact Method ESP $\lvert\mathscr{R}'\rvert$	t_{CPX}	t_{ESP}
P-n16-k8	450	450	449.0	99.8	1.0	2	1.0	443.7	98.6	444.7	98.8	450.0	100.0	0.8	0	0.0	0.8
P-n19-k2	212	212	212.0	100.0	1.0	–	1.0	202.7	95.6	212.0	100.0	212.0	100.0	0.8	0	0.0	0.8
P-n20-k2	216	216	213.0	98.6	1.0	–	1.0	209.8	97.1	215.2	99.6	216.0	100.0	2.2	0	0.0	2.2
P-n21-k2	211	211	211.0	100.0	1.0	–	1.0	209.4	99.2	211.0	100.0	211.0	100.0	1.2	0	0.0	1.2
P-n22-k2	216	216	216.0	100.0	2.0	–	2.0	212.9	98.6	215.5	99.8	216.0	100.0	3.0	0	0.0	3.1
P-n22-k8	603	603	603.0	100.0	3.0	2	3.0	596.0	98.8	603.0	100.0	603.0	100.0	12.3	0	0.0	12.3
P-n23-k8	529	529	529.0	100.0	18.0	2	18.0	528.0	99.8	529.0	100.0	529.0	100.0	0.2	0	0.0	0.2
P-n40-k5	458	458	456.9	99.8	28.0	–	34.0	456.2	99.6	456.8	99.7	458.0	100.0	26.2	0	0.0	26.2
P-n45-k5	510	510	506.6	99.3	59.0	3	194.0	500.6	98.2	505.0	99.0	509.8	100.0	21.5	0	0.0	21.5
P-n50-k7	554	554	551.5	99.5	79.0	3	143.0	548.0	98.9	548.8	99.1	554.0	100.0	16.7	0	0.0	16.7
P-n50-k8	649	631	616.3	97.7	102.0	3	9,272.0	611.8	97.0	615.5	97.5	624.7	99.0	11.8	182,191	585.0	596.8
P-n50-k10	696	696	689.3	99.0	50.0	3	304.0	684.1	98.3	687.5	98.8	695.4	99.9	9.5	1,904	0.9	10.4
P-n51-k10	741	741	735.2	99.2	35.0	3	105.0	729.4	98.4	733.0	98.9	741.0	100.0	9.4	0	0.0	9.4
P-n55-k7	568	568	557.9	98.2	90.0	2	4,649.0	553.7	97.5	557.5	98.2	563.2	99.2	28.5	24,701	177.1	205.6
P-n55-k8	588	588	579.8	98.6	42.0	2	1,822.0	571.9	97.3	578.6	98.4	583.7	99.3	28.3	13,434	46.2	74.5
P-n55-k10	699	694	681.4	98.2	107.0	3	9,076.0	677.2	97.6	680.7	98.1	689.3	99.3	15.9	24,612	50.5	66.4
P-n55-k15	993	989	972.8	98.4	251.0	3	1,944.0	959.8	97.0	968.9	98.0	986.1	99.7	8.3	3,235	0.6	8.9
P-n60-k10	756	744	738.9	99.3	126.0	3	570.0	734.3	98.7	737.8	99.2	743.4	99.9	13.4	54,810	46.0	59.4
P-n60-k15	1,033	968	962.8	99.5	118.0	3	442.0	955.5	98.7	960.9	99.3	968.0	100.0	0.0	0	0.0	0.0
P-n65-k10	792	792	786.0	99.2	159.0	3	422.0	781.2	98.6	785.3	99.2	792.0	100.0	0.0	0	0.0	0.0
P-n70-k10	834	827	814.5	98.5	292.0	3	24,039.0	808.3	97.7	811.0	98.1	821.7	99.4	35.1	72,927	739.1	774.2
P-n76-k4	593	593	588.8	99.3	363.0	–	572.0	586.2	98.8	587.6	99.1	590.3	99.5	84.8	488,946	2,126.3	2,211.1
P-n76-k5	627	627	616.8	98.4	273.0	–	14,546.0	612.3	97.7	615.3	98.1	619.5	98.8	122.1	>2,000,000	–	[a]
P-n101-k4	681	681	678.5	99.6	1,055.0	–	1,253.0	666.4	97.9	676.9	99.4	676.9	99.4	370.5	–	–	[b]
Averages				99.2	135.7		2,437.0		98.2		99.0		99.7	34.3			186.4

[a] CPLEX out of memory
[b] $|T| > NSTATB$ in procedure H^3 and in generating $\mathscr{R}'$

cannot be optimally solved by ESP as either CPLEX runs out of memory or the maximum size of the set T was reached in generating the final set of columns $\mathscr{R}'$.

On the instances solved to optimality by both methods, the new exact method ESP finds the optimal solutions in only a very small fraction of the computing time required by the branch-and-cut-and-price algorithm of Fukasawa et al.

Observe that instances P-n76-k4, P-n76-k5 and P-n101-k4 were solved by Fukasawa et al. using the branch-and-cut algorithm of Lysgaard et al. [32], rather than by branch-and-cut-and-price. Indeed, set partitioning based methods may not work well on loosely constrained instances (i.e., instances with large n and small k), since the number of promising routes can be huge in such cases. Thus, set partitioning based methods complement branch-and-cut approaches, which tend to work better on loosely constrained instances.

8 Conclusions

We have considered the CVRP, where a given fleet of delivery vehicles of identical capacity must service customers with known demands from a central depot at minimum routing cost. We have developed a new exact algorithm for the CVRP based on the set partitioning formulation with additional constraints that correspond to capacity and clique inequalities. The exact method uses a dual solution, computed by combining three different bounding procedures, to generate a reduced problem containing only the routes whose reduced costs are smaller than the gap between an upper bound and the lower bound achieved. The resulting problem is then solved by the integer programming solver CPLEX.

The computational results on CVRP instances taken from the literature show that the new lower bounds are better than the best lower bounds reported by Fukasawa et al. [21]. On the instances solved to optimality by both methods, the new exact method finds the optimal solutions in only a very small fraction of the computing time required by the procedure of Fukasawa et. al.

The proposed method can be easily adapted to deal with other routing constraints such as time windows, distance constraints, etc., simply by taking into account of such constraints in the route generation phase.

References

1. Augerat, P.: Approche polyédrale du problème de tournées de véhicules. Ph.D. thesis, Institut National Polytechnique de Grenoble (1995)
2. Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., Rinaldi, G.: Computational results with a branch and cut code for the capacitated vehicle routing problem. Tech. Rep. 1 RR949-M, ARTEMIS-IMAG, Grenoble France (1995)
3. Balas, E., Padberg, M.W.: Set partitioning: a survey. SIAM Rev. **18**, 710–760 (1976)
4. Baldacci, R., Bodin, L.D., Mingozzi, A.: The multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. Comput. Oper. Res. **33**, 2667–2702 (2006)
5. Baldacci, R., Hadjiconstantinou, E.A., Maniezzo, V., Mingozzi, A.: A new method for solving capacitated location problems based on a set partitioning approach. Comput. Oper. Res. **29**, 365–386 (2002)
6. Baldacci, R., Hadjiconstantinou, E.A., Mingozzi, A.: An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Oper. Res. **52**, 723–738 (2004)

7. Balinski, M., Quandt, R.: On an integer program for a delivery problem. Oper. Res. **12**, 300–304 (1964)
8. Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.): Network Routing, Handbooks in Operations Research and Management, vol. 8. Elsevier, Amsterdam (1995)
9. Barnhart, C., Laporte, G. (eds.): Transportation, Handbooks in Operations Research and Management Science, vol. 14. North-Holland, Amsterdam (2007)
10. Bianco, L., Mingozzi, A., Ricciardelli, S.: A set partitioning approach to the multiple depot vehicle scheduling problem. Optim. Methods Softw. **3**, 163–194 (1994)
11. Bodin, L.D., Mingozzi, A., Maniezzo, V. : Street routing and scheduling problems. In: Hall, R. (ed.) Handbook of Transportation Science, vol. 2, pp. 413–438. Kluwer, Boston (2003)
12. Bramel, J., Simchi-Levi, D.: Set-covering-based algorithms for the capacitated VRP. In: Toth, P., Vigo D. (eds.) The Vehicle Routing Problem, vol. 9, pp. 85–108 SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, (2002)
13. Christofides, N., Eilon, S.: An algorithm for the vehicle dispatching problem. Oper. Res. Q. **20**, 309–318 (1969)
14. Christofides, N., Mingozzi, A., Toth, P. : The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.) Combinatorial Optimization., pp. 315–338. Wiley, Chichester (1979)
15. Christofides, N., Mingozzi, A., Toth, P.: Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxation. Math. Program. **10**, 255–280 (1981)
16. Cordeau, J.-F., Laporte, G., Savelsbergh, M.W.P., Vigo, D. : Vehicle routing. In: Barnhart, C., Laporte, G. (eds.) Transportation, Handbooks in Operations Research and Management Science, vol. 14, pp. 367–428. North-Holland, Amsterdam (2007)
17. Cornuéjols, G., Harche, F.: Polyhedral study of the capacitated vehicle routing. Math. Program. **60**, 21–52 (1993)
18. CPLEX: ILOG CPLEX 9.0 callable library. ILOG (2004)
19. Fischetti, M., Toth, P.: An additive bounding procedure for combinatorial optimization problems. Oper. Res. **37**, 319–328 (1989)
20. Fisher, M.L.: Vehicle routing. In: Ball, M.O., Magnanti T.L., Monma, C.L., Nemhauser, G.L. Network Routing, Handbooks in Operations Research and Management Science, vol. 8, pp. 1–133. North-Holland, Amsterdam (1995)
21. Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragão, M., Reis, M., Uchoa, E., Werneck, R.F.: Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Math. Program. Ser. A **106**, 491–511 (2006)
22. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
23. Gendreau, M., Laporte, G., Potvin, J.-Y.: Metaheuristics for the capacitated VRP. In: Toth, P., Vigo D. (eds.) The Vehicle Routing Problem, vol. 9, pp. 129–154 SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, (2002)
24. Hoffman, K., Padberg, M.W.: Solving airline crew scheduling problems by branch and cut. Manage. Sci. **39**, 657–682 (1993)
25. Kim, S., Chang, K.N., Lee, J.Y.: A descent method with linear programming subproblems for nondifferentiable convex optimization. Math. Program. **71**, 17–28 (1995)
26. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. Eur. J. Oper. Res. **59**, 345–358 (1992)
27. Laporte, G., Nobert, Y., Desrochers, M.: Optimal routing under capacity and distance restrictions. Oper. Res. **33**, 1058–1073 (1985)
28. Laporte, G., Semet F.: Classical heuristics for the capacitated VRP. In: Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem, vol. 9, pp. 109–128. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)
29. Letchford, A.N., Eglese, R.W., Lysgaard, J.: Multistars, partial multistars and the capacitated vehicle routing problem. Math. Program. Ser. A **94**, 21–40 (2002)
30. Lübbecke, M.E., Desrosiers, J.: Selected topics in column generation. Oper. Res. **53**, 1007–1023 (2005)
31. Lysgaard, J.: CVRPSEP: A package of separation routines for the capacitated vehicle routing problem. Technical report, Department of Management Science and Logistics, Aarhus School of Business (2003)
32. Lysgaard, J., Letchford, A.N., Eglese, R.W.: A new branch-and-cut algorithm for the capacitated vehicle routing problem. Math. Program. Ser. A **100**, 423–445 (2004)

33. Marsten, R.E., Hogan, W.W., Blankenship, J.W.: The boxstep method for large-scale optimization. Oper. Res. **23**, 389–405 (1975)
34. du Merle, O., Villeneuve, D., Desrosiers, J., Hansen, P.: Stabilized column generation. Discrete Math. **194**, 229–237 (1999)
35. Mingozzi, A., Boschetti, M.A., Ricciardelli, S., Bianco, L.: A set partitioning approach to the crew scheduling problem. Oper. Res. **47**, 873–888 (1999)
36. Mingozzi, A., Christofides, N., Hadjiconstantinou, E.A.: An exact algorithm for the vehicle routing problem based on the set partitioning formulation. Tech. rep., University of Bologna (1994)
37. Mingozzi, A., Giorgi, S., Baldacci, R.: An exact method for the vehicle routing problem with backhauls. Transp. Sci. **33**, 315–329 (1999)
38. Naddef, D., Rinaldi, G.: Branch-and-cut algorithms for the capacitated VRP. In: Toth, P., Vigo D. (eds.) The Vehicle Routing Problem, vol. 9. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp. 53–81 (2002)
39. Niskanen, S., Östergård, P.R.J.: Cliquer user's guide. Tech. Rep. 48, Helsinki University of Technology Communications Laboratory (2003)
40. Östergård, P.R.J.: A fast algorithm for the maximum clique problem. Discrete Appl. Math. **120**, 197–207 (2002)
41. Rousseau, L.-M., Gendreau, M., Feillet, D.: Interior point stabilization for column generation. Oper. Res. Lett. (2006). doi:10.1016/j.orl.2006.11.004
42. Toth, P., Vigo, D.: Branch-and-bound algorithms for the capacitated VRP. In: Toth, P., Vigo D. (eds.) The Vehicle Routing Problem, vol. 9. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, pp. 29–51 (2002)
43. Toth, P., Vigo, D.: The Vehicle Routing Problem, vol. 9. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (2002)