CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Branch-and-Cut-and-Price Algorithm for the Capacitated Location-Routing Problem

Claudio Contardo
Jean-François Cordeau
Bernard Gendron

July 2011

CIRRELT-2011-44

Bureaux de Montréal :

Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :

Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Branch-and-Cut-and-Price Algorithm for the Capacitated Location-Routing Problem

Claudio Contardo[1,2,*], Jean-François Cordeau[1,3], Bernard Gendron[1,2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[3] Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. In this paper we present an exact algorithm for the Capacitated Location-Routing Problem (CLRP) based on column and cut generation. The CLRP is formulated as a set-partitioning problem which also inherits all of the known valid inequalities for the flow formulations of the CLRP. We introduce five new families of inequalities that are shown to dominate some of the cuts from the two-index formulation. The problem is then solved by column generation, where the sub-problem consists in finding a shortest path of minimum reduced cost under capacity constraints. We first use the two-index formulation for enumerating all of the possible subsets of depot locations that could lead to an optimal solution of cost smaller than or equal to a given upper bound. For each of these subsets, the corresponding Multiple Depot Vehicle Routing Problem is solved by means of column generation. The results show that we can improve the bounds found in the literature, solve to optimality some previously open instances, and improve the upper bounds on some other.

Keywords. Location-routing, vehicle routing, branch-and-cut-and-price, column generation.

Acknowledgements. The authors would like to thank Simon Spoorendonk and Enrico Bartolini for their insightful comments. Thanks are also due to the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds québécois de la recherche sur la nature et les technologies (FQRNT) for their financial support.

---

* Corresponding author: Claudio.Contardo@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2011

1. Introduction

In the Capacitated Location-Routing Problem (CLRP) we are given a set I of potential facilities and a set J of customers. With every facility $i \in I$ are associated a fixed opening cost f_i and a capacity b_i. To every customer $j \in J$ is associated a demand d_j. Distances are assumed to be symmetric. The problem can thus be defined on an undirected graph $G = (V, E)$, where $V = I \cup J$ is the vertex set and E is the edge set. To every edge $e = \{i, j\}$ we associate a routing cost c_{ij}. The fleet is assumed to be of unlimited size and homogeneous, each vehicle having a capacity Q. The objective is to choose a subset of facilities and to construct vehicle routes around these facilities to visit every customer exactly once, respecting both vehicle and facility capacities while minimizing the sum of fixed costs and routing costs.

The CLRP arises in several real-world applications. Labbé and Laporte (1986) solve the problem of locating postal boxes while minimizing a linear combination of routing costs (those of the mail collecting trucks) and customer inconvenience produced by their distance to the nearest postal box. Billionet et al. (2005) consider a location problem arising in mobile networks. The problem consists in locating radio-communication stations, designing rings and building antennaes inside these rings at minimum cost. Gunnarsson et al. (2006) solve a location-routing problem arising in the pulp distribution industry in Scandinavia.

The CLRP can be formulated as a three-index mixed-integer program (Perl and Daskin 1985). In such a formulation, asymmetries in the distance matrix and heterogeneities in the vehicle capacities can be easily taken into account. However, due to the large number of variables and its poor linear programming relaxation it has no practical use within an enumeration method such as branch-and-bound. In the context of exact algorithms for solving the CLRP, Belenguer et al. (2010) developed a two-index formulation and proposed several families of valid inequalities, such as y-Capacity Cuts (y-CC), Path Constraints (PC), Facility Degree Constraints (FDC), Imparity Constraints (IC) and Facility Capacity Inequalities (FCI). They solve the problem by means of branch-and-cut and their algorithm succeeds in solving small and medium size instances with up to 50 customers. Contardo et al. (2010) introduced three new formulations of the CLRP based on vehicle flows and commodity flows. They introduced strengthenings of the FCI as well as Location-Routing Comb Inequalities (LR-CI), Location-Routing Generalized Large Multistar Inequalities (LRGLM) and y-Generalized Large Multistar Inequalities (y-GLM), exploiting the fact that facilities have limited capacities. Their algorithms were able to solve instances containing up to 100 customers, the largest for branch-and-cut methods. Akca et al. (2009) developed a set-partitioning formulation based on a Dantzig-Wolfe decomposition of the three-index model. They solve the problem by means of branch-and-price, where the subproblem is a shortest path problem under capacity constraints (SPPRC). Their formulation provides reasonably good bounds at the root node of the search tree but does not appear to be effective for closing the gap using branching. Baldacci et al. (2010b) also formulate the CLRP as a set-partitioning problem. They use three different relaxations of the formulation that are applied sequentially in an additive manner. In the last step, they solve a small number of MDVRP by means of a cut-and-price-and-branch method, in which the root node is solved by colum generation, and then enumerate all of the remaining columns whose reduced cost is smaller than a given gap. The resulting integer program is then solved by means of a general-purpose integer programming solver. They use a stregthened version of the CC as well as clique inequalities. The bounds provided by their model are very tight, being able to solve instances with up to 199 customers and 14 facilities.

The CLRP is NP-hard as it generalizes both the Capacitated VRP (CVRP) and the Capacitated Facility Location Problem (CFLP). Moreover, the presence of capacities for both the vehicles and the facilities makes it particularly hard. Because of this, solution approaches for solving medium and large size instances have mainly focused on the development of heuristics. These heuristics in most cases use some decomposition scheme to divide the problem into a design sub-problem for the location decisions and an operational sub-problem for the routing part (Perl and Daskin 1985, Hansen et al. 1994, Wu et al. 2002, Lin et al. 2002, Liu and Lin 2005). Recently, Prins et al. (2006a,b, 2007) have proposed several metaheuristics that include memetic algorithms, cooperative Lagrangean relaxation with tabu search and greedy randomized adaptive search procedure (GRASP). Computational experience shows that the second approach is the most effective one for tackling large instances of the CLRP.

The contributions of this paper can be summarized as follows:

i. We adapt the set-partitioning formulation due to Akca et al. (2009) so that all of the cuts valid for the two-index formulation of the CLRP (Belenguer et al. 2010, Contardo et al. 2010) can be easily incorporated.

ii. We introduce two bounding procedures that are applied sequentially and that allow, in most cases, to reduce the CLRP to a series of multiple depot VRP, as in Baldacci et al. (2010b). Our computational results show that our bounding procedures can be stronger than those of Baldacci et al. (2010b) for some instances.

iii. We introduce several new families of cuts that are effective for closing the optimality gap. Moreover, our computational experience shows that using state-space relaxation in the pricing problem suffices to get bounds close to those obtained by pricing on elementary routes (routes that do not contain cycles).

iv. We introduce a new fathoming rule that accelerates the solution of the pricing subproblems.

As a result, our algorithm is able to solve all instances that are also solved by the exact method of Baldacci et al. (2010b) as well as four previously open instances. Additionally, we improve the best known feasible solution for three other instances. Moreover, for the instances that remain unsolved we improve the best known lower bounds.

The paper is organized as follows. In Section 2 we present some formulations of the CLRP, namely the two-index vehicle-flow formulation due to Belenguer et al. (2010) as well as the set-partitiong formulation due to Akca et al. (2009). In Section 3 we describe the valid inequalities used through this paper. It includes some known valid inequalities from the two-index formulation and the set-partitioning problem as well as new valid inequalities that are shown to be valid for the set-partitioning formulation of the CLRP. In Section 4 we describe the exact algorithm used to solve the CLRP to optimality. We first describe the separation algorithms used to find violated valid inequalities. We then describe the different bounding procedures as well as the pricing algorithms used to solve the corresponding set-partitioning problems. Finally, we discuss some computational issues that are mostly implementation-specific and that have an important impact on the performance of the algorithm. In Section 5 we present our computational results and compare against the state-of-the-art solvers for solving the CLRP. We conclude in Section 6 with a summary of the proposed methodology and discuss possible avenues of future research.

2. CLRP Formulations

In this section we first present the two-index vehicle-flow formulation of the CLRP due to Belenguer et al. (2010) and the set-partitioning formulation introduced by Akca et al. (2009). We also show that any inequality valid for the two-index formulation can be easily extended to the set-partitioning formulation.

2.1. Two-index vehicle-flow formulation

Belenguer et al. (2010) proposed the following two-index vehicle-flow formulation for the CLRP. For every vertex set U, let $\delta(U)$ be the edge subset containing all those edges with exactly one endpoint in U. For two disjoint vertex sets T, U, let $(T : U)$ be the edge subset containing all edges with one endpoint in T and the other in U. For every facility $i \in I$, let z_i be a binary variable equal to 1 iff facility i is selected for opening. For every edge $e \in E$, let x_e be a binary variable equal to 1 iff edge e is traversed once by some vehicle. Finally, for every edge $e \in \delta(I)$ let y_e be a binary variable equal to 1 iff edge e is used twice by some vehicle. For a given edge set $F \subseteq E$ let $x(F) = \sum_{e \in F} x_e$, $y(F) = \sum_{e \in F} y_e$. For a given customer subset $S \subseteq J$, let $d(S) = \sum_{j \in S} d_j$ and $r(S) = \lceil d(S)/Q \rceil$ (which actually is a lower bound on the number of vehicles needed to serve the customers in S). The formulation is the following.

$$\min \quad \sum_{i \in I} f_i z_i + \sum_{e \in E} c_e x_e + 2 \sum_{e \in \delta(I)} c_e y_e \tag{TIF}$$

subject to

$$
\begin{aligned}
& x(\delta(j)) + 2y(I : \{j\}) = 2 && j \in J && (1)\\
& x(\delta(S)) + 2y(I : S) \geq 2r(S) && S \subseteq J, |S| \geq 2 && (2)\\
& x_{ij} + y_{ij} \leq z_i && i \in I, j \in J && (3)\\
& x(I : \{j\}) + y(I : \{j\}) \leq 1 && j \in J && (4)\\
& x((I \setminus \{i\}) \cup \overline{S} : S) + 2y(I \setminus \{i\} : S) \geq 2 && i \in I, S \subseteq J, d(S) > b_i && (5)\\
& x(\delta(S)) \geq 2(x(\{h\} : I') + x(\{j\} : I \setminus I')) && S \subseteq J, |S| \geq 2, h, j \in S, I' \subset I && (6)\\
& z_i \in [0, 1] \text{ and integer} && i \in I && (7)\\
& x_e \geq 0 \text{ and integer} && e \in E && (8)\\
& y_e \geq 0 \text{ and integer} && e \in \delta(I). && (9)
\end{aligned}
$$

Constraints (1) are the degree constraints at customer nodes. Constraints (2) are capacity cuts (CC), whose role is to forbid at the same time proper tours disconnected from facilities and tours serving a demand larger than Q. Constraints (3) ensure that there is no outgoing flow leaving from closed facilities. Constraints (4) are the path constraints for single customers. They forbid routes of the form $i_1 \to j \to i_2, i_1, i_2 \in I, i_1 \neq i_2, j \in J$. Constraints (5) are the facility capacity inequalities (FCI). They forbid the existence of routes leaving from a same facility i and serving a demand larger than b_i. Constraints (6) are the path constraints (PC) for multiple customers. Their role is to prevent the route of a single vehicle from joining two different facilities.

Belenguer et al. (2010) have shown that constraints (2) can be strengthened into the so-called y-Capacity Cuts (y-CC):

$$x(\delta(S)) + 2y(I : S \setminus S') \geq 2r(S) \qquad S \subseteq J, |S| \geq 2, S' \subset S, r(S \setminus S') = r(S). \tag{10}$$

These authors showed that the FCI can be generalized to take into account several facilities in the same constraint. For a subset $I' \subseteq I$ of facilities, they define $r(S, I') = \lceil (d(S) - b(I'))/Q \rceil$ (which is a lower bound on the number of vehicles that are needed to serve the demand of customers in S from facilities other than those in I'), where $b(I') = \sum_{i \in I'} b_i$. The following constraint, introduced by Contardo et al. (2010) and called strengthened FCI (SFCI), takes into account this observation and can be shown to dominate the FCI as well as the SFCI introduced by Belenguer et al. (2010):

$$x(I \setminus I' : S) + 2y(I \setminus I' : S \setminus S') \geq 2r(S, I') \qquad S \subseteq J, I' \subseteq I, S \subset S', r(S \setminus S', I') = r(S, I'). \tag{11}$$

2.2. Set-partitioning formulation

We now describe the set-partitioning formulation introduced by Akca et al. (2009) and also used by Baldacci et al. (2010b), and link it to the two-index vehicle-flow formulation so that all of the known cuts for the CLRP are also valid. Let us denote by Ω_i the set of all routes (possibly containing cycles) starting and ending at facility $i \in I$ and servicing a subset of at least two customers with total demand of Q or less, and let $\Omega = \cup_{i \in I} \Omega_i$ be the set of all possible routes servicing two or more customers with total accumulated demand of Q or less. For every $l \in \Omega$ let us associate a binary variable λ_l equal to 1 if l appears in the optimal solution of the CLRP and 0 otherwise, and a cost c_l for using this route. For every edge $e \in E$ and route $l \in \Omega$ let q_l^e be the number of times that edge e appears in route l. If Ω is restricted to contain only elementary routes then q_l^e is a binary constant, otherwise it can be a general integer. On the other hand, let us define binary variables y_{ij} for $\{i, j\} \in \delta(I)$ equal to 1 iff customer j is served from facility i by a single-customer route. Note that if distances satisfy the triangular inequality, the optimal solution of this problem will only contain elementary paths even if Ω is enlarged to contain routes with cycles. In fact, in this case it is always possible to build from a solution with cycles, another solution with elementary routes at lower cost. Let us extend the demands to facility nodes by letting $d_v = 0$ for every $v \in I$. A valid formulation for the CLRP is

$$\min \quad \sum_{i\in I} f_i z_i + \sum_{l\in\Omega} c_l\lambda_l + 2\sum_{e\in\delta(I)} c_e y_e \tag{SPF}$$

subject to

$$\sum_{l\in\Omega}\sum_{e\in\delta(\{j\})} q_l^e\lambda_l + 2y(I:\{j\}) = 2 \qquad j\in J \tag{12}$$

$$\sum_{l\in\Omega_i}\sum_{e=\{h,j\}\in E}(d_h+d_j)q_l^e\lambda_l + 2\sum_{e\in\delta(\{i\})} d_j y_{ij} \le 2b_i z_i \qquad i\in I \tag{13}$$

$$\lambda_l \ge 0 \text{ and integer} \qquad l\in\Omega \tag{14}$$

$$y_e \ge 0 \text{ and integer} \qquad e\in\delta(I) \tag{15}$$

$$z_i \in [0,1] \text{ and integer} \qquad i\in I \tag{16}$$

In this formulation, constraints (12) ensure that each customer is served exactly once. Constraints (13) are the facility capacity inequalities. They ensure that the demand served from any facility i will not exceed its capacity b_i. The distinction between single-customer and multiple-customer routes naturally defines a relationship between vehicle-flow variables x from the two-index formulation and λ, as follows

$$\sum_{l\in\Omega} q_l^e\lambda_l - x_e = 0 \qquad e\in E \tag{17}$$

In such a way, all of the valid inequalities from the two-index formulation of the CLRP can be translated into the set-partitioning formulation by using identities (17).

3. Valid inequalities

In this section we describe the valid inequalities that can be applied to formulation (SPF) and that strengthen the LP relaxation. First, we describe some of the valid inequalities that have been developed in the context of the two-index and three-index formulations by Belenguer et al. (2010) and Contardo et al. (2010). We then describe new families of valid inequalities that are shown to dominate several of the former and that effectively strengthen formulation (SPF).

3.1. Valid inequalities for the two-index formulation

The valid inequalities for formulation (TIF) include several different families. After a series of preliminary tests, we have decided to keep only a subset of them, namely strengthened comb inequalities (SCI), framed capacity inequalities (FrCI), effective strengthened facility capacity inequalities (ESFCI), facility degree constraints (FDC) and location-routing comb inequalities (LR-CI). To include any of these constraints into formulation (SPF) we use identity (17). For details on the inequalities we refer to Lysgaard et al. (2004), Belenguer et al. (2010) and Contardo et al. (2010).

3.2. Valid inequalities for the set-partitioning formulation

The valid inequalities for the set-partitioning formulation include a strengthening of the y-SCC introduced by Baldacci et al. (2008) for solving the CVRP and also used by Baldacci et al. (2010b) for the CLRP. We also introduce strengthenings of the degree constraints (12), of SFCI constraints (11), ESFCI and FrCI. We complement this with the addition of subset-row inequalities (SRI).

Any constraint in the two-index space can be translated into a constraint in the route space by using identity (17). However, the constraints translated this way will not take into account the fact that a route can cross more than once a given subset of vertices. For a given subset of routes $\mathcal{R} \subseteq \Omega$, let us define $\lambda(\mathcal{R}) = \sum_{l \in \mathcal{R}} \lambda_l$. We also let $i(l)$, $J(l)$ and $E(l)$ to be the facility to which l is assigned, the set of customers served by l and the set of edges used by l, respectively.

3.2.1. y-Strengthened CC (y-SCC)

Let us consider, for a given customer set $S \subseteq J$ and subset $S' \subset S$ such that $r(S \setminus S') = r(S)$, the corresponding y-CC as described in Belenguer et al. (2010) and Contardo et al. (2010), for formulations (TIF) and (SPF), respectively:

$$x(\delta(S)) + 2y(I : S \setminus S') \geq 2r(S) \qquad \text{for TIF} \tag{18}$$

$$\sum_{l \in \Omega} \sum_{e \in \delta(S)} q_l^e \lambda_l + 2 \sum_{e \in [I:S \setminus S']} y_e \geq 2r(S) \qquad \text{for SPF.} \tag{19}$$

Baldacci et al. (2008) noted that the CC (2) can be strengthened by setting the coefficient of a given path variable λ_l to be 0 if l does not serve a customer in S and 1 otherwise, rather than counting the number of edges of l that are also in $\delta(S)$. For formulation (SPF), the constraint is the following,

$$\lambda(\{l : J(l) \cap S \neq \emptyset\}) + y(I : S) \geq r(S). \tag{20}$$

For the y-CC we can apply the same reasoning, as stated in the following proposition.

Proposition 3.1. *Let $S \subseteq J$ be a subset of customers, and $S' \subset S$ such that $r(S \setminus S') = r(S)$, the following constraint is valid for the CLRP and dominates the y-CC* (19) *and the SCC* (20)*.*

$$\lambda(\{l : J(l) \cap S \neq \emptyset\}) + y(I : S \setminus S') \geq r(S). \tag{21}$$

We call this constraint the y-Strengthened CC (y-SCC).

Proof Let us define $\mathcal{L}(T) = \{l \in \Omega : J(l) \cap T \neq \emptyset\}$, for $T \subseteq J$. We then have

$$\lambda(\mathcal{L}(S)) = \lambda(\mathcal{L}(S \setminus S')) + \lambda(\mathcal{L}(S')) - \lambda(\mathcal{L}(S \setminus S') \cap \mathcal{L}(S')). \tag{22}$$

We have to prove that

$$\lambda(\mathcal{L}(S)) + y(I : S \setminus S') \geq r(S)$$

is a valid inequality of the CLRP. In fact, we have

$$\begin{aligned}\lambda(\mathcal{L}(S)) + y(I : S \setminus S') &= \lambda(\mathcal{L}(S \setminus S')) + \lambda(\mathcal{L}(S')) - \lambda(\mathcal{L}(S \setminus S') \cap \mathcal{L}(S')) + y(I : S \setminus S') \\ &\geq r(S \setminus S') + \lambda(\mathcal{L}(S')) - \lambda(\mathcal{L}(S \setminus S') \cap \mathcal{L}(S')) \\ &\geq r(S \setminus S') \\ &= r(S).\end{aligned}$$

The dominance with respect to the y-CC comes from the fact that a route l that visits a customer set S must have two or more edges crossing it, and the dominance with respect to the SCC comes from the consideration of the customer set S'. □

3.2.2. Strengthened Degree Constraints (SDEG)

Degree constraints in the two-index space count the number of times that a certain node is traversed. If a node can be traversed several times by a single route, then a stronger version of the degree constraint is

$$\lambda(\{l \in \Omega : j \in J(l)\}) + y(I : \{j\}) \geq 1 \qquad j \in J. \tag{23}$$

These constraints are relevant when, instead of restricting the state-space to elementary routes, it is rather relaxed to contain routes with cycles. In our algorithm we have found that the addition of these constraints when pricing on non-elementary routes is an effective method to get bounds close to the ones obtained by pricing on elementary routes. Indeed, the problem of finding an appropriate balance between speed and lower bound quality for different variants of the SPPRC has already been studied and is a key aspect in the performance of column generation based algorithms for vehicle routing problems (see, e.g., Boland et al. 2006, Righini and Salani 2008, Desaulniers et al. 2008). This intuition is supported by the following proposition,

Proposition 3.2. *The optimal value of the linear relaxation of* (SPF) *when restricting the space* Ω *to elementary routes is the same as when* Ω *is enlarged to routes with cycles after adding the SDEG constraints* (23)*.*

Proof Obviously elementary routes satisfy constraints (23), so the value of the linear relaxation on the elementary case is at least as good as in the relaxed case. On the other hand, in the relaxed case, no route with cycles will be basic after the addition of (23). Indeed, let $j \in J$ be any customer, and let $\Omega_j, \Omega_{cyc(j)}$ be the subsets of routes traversing j and containing a cycle in j, respectively (obviously $\Omega_{cyc(j)} \subseteq \Omega_j$). For that customer, from constraints (12) we have

$$\sum_{l \in \Omega_j \setminus \Omega_{cyc(j)}} \sum_{e \in \delta(\{j\})} q_l^e \lambda_l = 2 - 2y(I : \{j\}) - \sum_{l \in \Omega_{cyc(j)}} \sum_{e \in \delta(\{j\})} q_l^e \lambda_l \tag{24}$$

Using (23), $\Omega_{cyc(j)}$ also has to satisfy

$$\sum_{l \in \Omega_j \setminus \Omega_{cyc(j)}} \lambda_l \geq 1 - y(I : \{j\}) - \sum_{l \in \Omega_{cyc(j)}} \lambda_l \tag{25}$$

After multiplying the second equation by two, the left-hand side of both equations coincide, and the following relationship holds between their right-hand sides

$$\sum_{l \in \Omega_{cyc(j)}} \sum_{e \in \delta(\{j\})} q_l^e \lambda_l \leq \sum_{l \in \Omega_{cyc(j)}} 2\lambda_l \tag{26}$$

As $\sum_{e \in \delta(\{j\})} q_l^e \geq 4$ for $l \in \Omega_{cyc(j)}$ (because j is traversed at least twice, i.e. by at least 4 edges), it follows that $\lambda_l = 0$ for every $l \in \Omega_{cyc(j)}$. □

3.2.3. Set-Partitioning SFCI (SP-SFCI)

Let us consider the SFCI constraints (11), and let S, I' and S' be as in (11). The following strengthening of the SFCI, called Set-Partitioning SFCI (SP-SFCI), is valid for the CLRP and dominates (11):

$$\sum_{k \in I \setminus I'} \lambda(\{l \in \Omega_k : J(l) \cap S \neq \emptyset\}) + y(I \setminus I' : S \setminus S') \geq r(S, I'). \tag{27}$$

Before proving the validity of the above constraint, let us define some notation. For each $i \in I, j \in S$, let w_{ij} be a binary constant equal to 1 iff customer j is served from facility i. Let $W_{I'} = \{j \in J : w_{ij} = 1 \text{ for some } i \in I'\}$. For given subsets $H \subseteq I$ and $S \subseteq J$, let us define $\mathcal{L}_H(S) = \cup_{i \in H}\{l \in \Omega_i : J(l) \cap S \neq \emptyset\}$. Now, let us prove the validity of constraints (27).

Proposition 3.3. *Constraints* (27) *are valid for the CLRP and dominate the SFCI* (11).

Proof Let us consider first the case $S' = \emptyset$. Indeed, if $S \subseteq W_{I'}$ then constraint (27) is trivially satisfied (because $r(S, I') = 0$). If $S \subseteq \overline{W}_{I'}$ then $\lambda(\mathcal{L}_{I'}(S)) = y(I' : S) = 0$ and therefore $\lambda(\mathcal{L}_{I \setminus I'}(S)) + y(I \setminus I' : S) = \lambda(\mathcal{L}_I(S)) + y(I : S) \geq r(S) \geq r(S, I')$. If $S \cap W_{I'} \neq \emptyset$ and $S \cap \overline{W}_{I'} \neq \emptyset$, we have $\lambda(\mathcal{L}_{I \setminus I'}(S)) + y(I \setminus I' : S) = \lambda(\mathcal{L}_{I \setminus I'}(S \cap W_{I'})) + y(I \setminus I' : S \cap W_{I'}) + \lambda(\mathcal{L}_{I \setminus I'}(S \cap \overline{W}_{I'})) + y(I \setminus I' : S \cap \overline{W}_{I'}) \geq r(S \cap W_{I'}, I') + r(S \cap \overline{W}_{I'}) \geq r(S, I')$. Let us suppose now that $S' \neq \emptyset$. Let $S'' \subseteq S'$ be such that $y(I \setminus I' : S') = |S''|$, i.e., the customers in S' that are served by single-customer routes from facilities in $I \setminus I'$ are exactly those in S''. As a consequence of this, $\lambda(\mathcal{L}_{I \setminus I'}(S)) = \lambda(\mathcal{L}_{I \setminus I'}(S \setminus S''))$ and then $\lambda(\mathcal{L}_{I \setminus I'}(S)) + y(I \setminus I' : S \setminus S') = \lambda(\mathcal{L}_{I \setminus I'}(S \setminus S'')) + y(I \setminus I' : S \setminus S'') \geq r(S \setminus S'', I') = r(S, I')$. The dominance with respect to constraints (11) comes from the fact that *i*) routes crossing set S several times are only counted once, and *ii*) edges connecting S with $\overline{S}$ are considered only if they belong to routes departing from $I \setminus I'$. □

3.2.4. Set-Partitioning ESFCI (SP-ESFCI)

The Effective SFCI were introduced by Belenguer et al. (2010) and Contardo et al. (2010) and are valid for the two-index formulation (TIF). They can be seen as a strengthening of the SFCI by noticing that the right-hand side of such constraint can be in fact lifted whenever $z_i = 0$ for some $i \in I'$. For the set-partitioning formulation (SPF) they can be written as

$$\sum_{k \in I \setminus I'} \lambda(\{l \in \Omega_k : J(l) \cap S \neq \emptyset\}) + y(I \setminus I' : S \setminus S') \geq r(S, I') + z_i(r(S, I' \setminus \{i\}) - r(S, I')). \tag{28}$$

The validity proof follows from the validity of the SP-SFCI for the two cases $z_i = 1$ and $z_i = 0$.

3.2.5. Strengthened Framed Capacity Inequalities (SFrCI)

The framed capacity inequalities were developed by Augerat (1995) for the CVRP and later succesfully used by other authors in the development of algorithms based on cutting planes and column generation (Lysgaard et al. 2004, Fukasawa et al. 2006). Given a customer set S, that we call the frame, and a partition of it $(S_i)_{i=1}^t$, the related FrCI seen in formulation (TIF) is

$$x(\delta(S)) + 2y(I : S) + \sum_{i=1}^{t}(x(\delta(S_i)) + 2y(I : S_i)) \geq 2\left(BPP(S|(S_i)_{i=1}^t) + \sum_{i=1}^{t} r(S_i)\right), \quad (29)$$

where $BPP(S|(S_i)_{i=1}^t)$ represents the solution of the following bin-packing problem. For every $i = 1, \ldots, t$ consider $\lceil d(S_i)/Q \rceil$ items of size Q except for the last item that will have size $d(S_i) - (\lceil d(S_i)/Q \rceil - 1)Q$. Also, set the bins to have size Q. In addition to using identity (17) to adapt this constraint to formulation (SPF), the same observation as done for the y-SCC, SDEG, SP-SFCI and SP-ESFCI can be applied. The following constraint, called strengthened FrCI (SFrCI) is valid for the CLRP and also dominates the FrCI.

$$\lambda(\{l \in \Omega : J(l) \cap S \neq \emptyset\}) + \sum_{i=1}^{t} \lambda(\{l \in \Omega : J(l) \cap S_i \neq \emptyset\}) + 2y(I : S) \geq$$
$$BPP(S|(S_i)_{i=1}^t) + \sum_{i=1}^{t} r(S_i). \quad (30)$$

Before proving the validity of the SFrCI we need the following lemma

Lemma 3.4 (Augerat (1995)). *Let $S \subseteq J$ and $(S_i)_{i=1}^t$ a partition of S. If $\lceil d(S_1 \cup S_2)/Q \rceil = \lceil d(S_1)/Q \rceil + \lceil d(S_2)/Q \rceil$ then $BPP(S|S_1, S_2, \ldots, S_t) \geq BPP(S|S_1 \cup S_2, S_3, \ldots, S_t)$. Otherwise $BPP(S|S_1, S_2, \ldots, S_t) + 1 \geq BPP(S|S_1 \cup S_2, S_3, \ldots, S_t)$.*

Proof See Augerat (1995). □

Proposition 3.5. *Constraints* (30) *are valid for* (SPF).

Proof The proof uses exactly the same arguments as in Augerat (1995). Let us suppose first that sets S_i satisfy $d(S_i) \leq Q$. Let us consider the bin-packing problem defined above, with objects of sizes $d(S_i)$ for every $i = 1, \ldots, t$ and bin size equal to Q. Let us denote the set of objects by K. In this context, let us call a *cut of object k in K* the following operation: remove k (of size $d(k)$) from K and replace it by two smaller objects whose total size is equal to $d(k)$. It is known that after a cut operation, the solution of a BPP is reduced by at most one unit. As a consequence, the same applies for q cut operations, so that the solution of the BPP is reduced by at most q units. In the case of the CLRP, the quantity $w = \sum_{i=1}^{t}(\lambda(\{l \in \Omega : J(l) \cap S_i \neq \emptyset\}) + y(I : S_i) - 1)$ represents exactly the number of cuts that are applied to the set S, and thus $BPP(S|(S_i)_{i=1}^{t}) + w$ represents a lower bound on the number of vehicles needed to serve the demand of S. Now, in the general case, let (λ, y, z) be a solution of (SPF). For every subset S_i, (λ, y) define a partition $S_i^k, k = 1, \dots n_i$ of subsets of S_i such that *i)* $\lambda(\{l \in \Omega : J(l) \cap S_i^k \neq \emptyset\}) + y(I : S_i^k) = 1$ and *ii)* $n_i = \lambda(\{l \in \Omega : J(l) \cap S_i \neq \emptyset\}) + y(I : S_i)$. From the first case we have that $\lambda(\{l \in \Omega : J(l) \cap S \neq \emptyset\}) + y(I : S) \geq BPP(S|(S_1^k)_{k=1}^{n_1}, (S_2^k)_{k=1}^{n_2}, \dots, (S_t^k)_{k=1}^{n_t})$. For every $i = 1, \dots, t$ we apply n_i successive contractions of the subsets S_i^k and compute $\alpha(i, j)$ equal to the number of times that $BPP(S|(S_1^k)_{k=1}^{n_1}, (S_2^k)_{k=1}^{n_2}, \dots, (S_t^k)_{k=1}^{n_t})$ decreases by one unit after a contraction. By applying the lemma, we have that $\alpha(i, 1) = \lceil d(S_i^1)/Q \rceil + \lceil d(S_i^2)/Q \rceil - \lceil d(S_i^1 \cup S_i^2)/Q \rceil = 2 - \lceil d(S_i^1 \cup S_i^2)/Q \rceil$ and, more generally, $\alpha(i, j) = j + 1 - \lceil d(\bigcup_{k=1}^{j} S_i^k)/Q \rceil$. At the end of all of these successive contractions we will have that $\lambda(\{l \in \Omega : J(l) \cap S \neq \emptyset\}) + y(I : S) \geq BPP(S|(S_i)_{i=1}^{t}) - \sum_{i=1}^{t}(n_i - \lceil d(S_i)/Q \rceil)$ □

3.2.6. Subset-Row Inequalities (SRI, Jepsen et al. (2008))

The subset-row inequalities are a special case of the clique inequalities (Balas and Padberg 1976) and are valid for the set partitioning formulation of the CLRP. Let us consider the conflict graph $\mathcal{H}_\lambda$ constructed as follows. The vertices of $\mathcal{H}_\lambda$ are the routes $l \in \Omega$ such that $\lambda_l > 0$. Two vertices in $V(\mathcal{H}_\lambda)$ are linked by an edge if they share at least one customer. A clique in $\mathcal{H}_\lambda$ is a maximal complete induced subgraph of $\mathcal{H}_\lambda$. For every clique $\mathcal{C} \subseteq \mathcal{H}_\lambda$, the following clique inequality is valid for the CLRP:

$$\sum_{v \in V(\mathcal{C})} \lambda_v \leq 1. \tag{31}$$

The addition of clique inequalities into the master problem SPF has, however, an important drawback: they make the pricing problem of finding routes (with or without cycles) of negative reduced cost much more difficult. Indeed, during the pricing problem it must be checked if a partial path participates or not in a clique. This is equivalent to checking if a partial path intersects every column already in a clique in at least one customer node, which in practice is difficult to do. Jepsen et al. (2008) introduced the subset-row inequalities. A subset-row inequality is a clique inequality associated to a clique $\mathcal{C}$ to which we assign a subset of customers $\chi(\mathcal{C}) \subseteq J$ such that every column in $\mathcal{C}$ intersects $\chi(\mathcal{C})$ in at least a certain number of customers. If $|\chi(\mathcal{C})|$ is small, the pricing problem can be accelerated as only $|\chi(\mathcal{C})|$ comparisons are needed to check if a given path participates in the clique. These inequalities are a particular case of the clique inequalities and in general provide slightly weaker bounds. The results obtained by Jepsen et al. (2008) for the particular case of $|\chi(\mathcal{C})| = 3$ show that the gain for considering the clique inequalities instead of the subset-row inequalities is usually not worth the extra computational effort.

4. Solution Methodology

In this section we describe the exact algorithm that solves the CLRP to optimality. We first describe the separation algorithms used in order to find violated inequalities. Then, we describe two bounding procedures that are applied sequentially. The first procedure is based on the two-index formulaton (TIF) with additional cuts. The second procedure is based on the set partitioning formulation (SPF) with additional cuts. We then describe an enumeration procedure to close the optimality gap that is applied only in certain cases. Finally, we describe the computational issues in the implementation of the proposed algorithm.

4.1. Separation Algorithms

We now describe the separation algorithms used to separate the different families of valid inequalities used in our algorithm. Our separation strategy is as follows: we first try to generate cuts translated from the two-index formulation (TIF). If no such cuts can be found, we try to generate cuts SDEG, y-SCC, SP-SFCI, SP-ESFCI and SFrCI. If it fails, we try to generate cuts SRI. This strategy allows us to keep the number of strong constraints small as their inclusion in the pricing algorithm make it harder.

4.1.1. Inequalities translated from formulation *TIF*

For the valid inequalities translated from the two-index formulation using identity (17), such as y-CC, SFCI, ESFCI, SCI, LR-CI or FrCI, we use the separation algorithms introduced by Lysgaard et al. (2004), Belenguer et al. (2010) and Contardo et al. (2010).

4.1.2. SDEG, y-SCC, SP-SFCI, SP-ESFCI and SFrCI

Although there is a polynomial number of SDEG constraints, we do not add them all at the beginning of the algorithm, but we rather check if for a certain weak degree constraint, its related strong constraint is violated, and add it to the problem. For the remaining constraints, we use the same principle. In fact, we check if, for any previously found weak constraint y-CC, SFCI, ESFCI or FrCI, its related strong constraint is violated and in this case we add it to formulation SPF.

4.1.3. Subset-Row Inequalities

The separation of the subset-row inequalities is done by enumeration just as in Jepsen et al. (2008). Indeed, we only separate SRI for cliques $\mathcal{C}$ such that $|\chi(C)| = 3$. We check for every triplet $(i, j, k) \in J^3$, $i < j < k$ if the corresponding SRI is violated. If it is the case, it is added to the master problem.

4.2. First bounding procedure

In this procedure, an enumeration method based on a branch-and-cut algorithm (Contardo et al. 2010) is applied to problem (TIF) after dropping the integrality constraints on the edge variables x and y. This procedure is used to obtain candidate subsets $I' \subseteq I$ of facilities such that the problem restricted to these facilities could lead to a feasible solution with cost smaller or equal than a given upper bound. We denote the set that contains the subsets I' by $\mathcal{I}$. For finding the subsets in $\mathcal{I}$, a good upper bound is needed to prune nodes in the branching tree. In our method, we have used the best feasible solutions found in the literature. For large instances, however, the computation of the whole branching tree can be prohibitive. In this case, the branch-and-bound algorithm is terminated earlier and the uninspected nodes are also added to $\mathcal{I}$. Now, the facilities in a given subset $I' \in \mathcal{I}$ are not only those that are open but also those that could not be fixed in the current node. During the process, different families of valid inequalities are added to strengthen the formulation. However, we only add cuts in nodes whose depth is less than or equal to 5. For each candidate set $I' \subseteq I$ generated by the algorithm we proceed as follows:

i. Based on reduced costs, perform variable fixing on the location variables z, in case set I' contains facilities that remained unfixed.

ii. Based on reduced costs, perform variable fixing on the edge variables x.

iii. Compute the optimal dual variables associated to the degree constraints (1).

iv. Compute $K_m(I')$ as an upper bound on the maximum number of routes that serve two or more customers, namely $K_m(I') = \lfloor \max\{\frac{1}{2}x(\delta(I')) : (x, y, z) \in \mathcal{A}\} \rfloor$, where $\mathcal{A}$ stands for the set of constraints (1)-(9) plus the generated cuts and after dropping the integrality conditions.

For each subset I' found by this algorithm we apply a second bounding procedure and a column enumeration method (in this context, the definition of set I' is implicit and will sometimes be omitted). Note that Baldacci et al. (2010b) use a similar approach, except that their first bounding procedure computes a global lower bound obtained by solving a relaxation of the set-partitioning problem. This bound is then used to discard non promising subsets $I' \subseteq I$. In Section 5 we presentr computational results comparing the first bounding procedure that we propose with the one suggested by Baldacci et al. (2010b).

4.3. Second bounding procedure

In this procedure, the following state-space relaxation of formulation (SPF) is solved by means of column generation. Instead of considering elementary routes (i.e., routes without cycles), we allow routes that contain cycles of length three or more, i.e., for nodes $i \neq j \neq k \neq i$ the subpaths $i \to i$, $i \to j \to i$ are forbidden, but the sequence $i \to j \to k \to i$ is permitted. The pricing problem consists in finding routes without cycles of length one or two and such that the reduced costs are minimized. This problem is known in the literature as the 2-cyc-SPPRC (Desrochers et al. 1992). This is an important difference with respect to the method of Baldacci et al. (2010b) in which the resolution of the subproblem is restricted to elementary routes. During the computation, we add the cuts described in Section 3. The violation threshold for the strong cuts is initially set to 0.3. When no more columns of negative reduced cost or violated cuts can be detected, the current objective function value is in fact a valid lower bound for the problem. Let us call this lower bound z^*. We run algorithm ENUM-ESPPRC (described in the next section) in order to price out the remaining columns $l \in \Omega$ such that $\overline{c}_l \leq z_{UB} - z^*$. We have set two hard limits to algorithm ENUM-ESPPRC: the number of labels cannot exceed at any time a maximum $\phi_{max} = 10^6$, and the total number of generated columns cannot exceed $\Delta_{max} = 10^7$. In case of success of this procedure, the columns generated are stored in a column pool $\mathcal{P}$ and the violation threshold for strong constraints is lowered to 0.01. Otherwise, we lower the violation threshold (thus generating more cuts) and continue with the process. This is done at most three times before finishing the column generation process. For instance, for the case of constraints SDEG, the sequence of violation thresholds is $(0.3, 0.25, 0.2, 0.1)$. Whenever the column enumeration ENUM-ESPPRC is done with success, at every following iteration of the column generation method, we do not solve the pricing problem 2-cyc-SPPRC but rather check the reduced costs of columns in $\mathcal{P}$. Note that the size of set $\mathcal{P}$ can be huge and computing the reduced cost of every column in it can be very cumbersome. For dealing with this issue, at every iteration after the creation of $\mathcal{P}$ in which no columns of negative reduced cost were found, we also delete from the pool all the columns l such that $\overline{c}_l > z_{UB} - z^*$. At the very end of the bounding procedure, we either prune the current node if the final lower bound is greater than or equal to z_{UB}, or otherwise solve the integer problem with the columns generated so far, with the hope of improving the upper bound. In what follows, we first describe the decomposition of the reduced costs for the constraints translated from formulation (TIF), namely all of the constraints in (SPF) plus the cuts that are valid for this formulation. We then show how to incorporate the set-partitioning constraints, such as y-SCC, SDEG, SP-SFCI, SP-ESFCI, SFrCI and SRI into the computation of the reduced costs. We then describe the pricing problem 2-cyc-SPPRC that suits our problem with the additional cuts. We end by describing how we compute lower bounds out of the result of the pricing problem.

4.3.1. Decomposition of the reduced costs edge-by-edge

Let us first suppose that only constraints (12), (13) (with duals α and β, respectively) have been added to the problem. For every $i \in I'$, define the reduced cost of an edge $e \in E(J) \cup \delta(\{i\})$ as

$$\overline{c}_e = \begin{cases} c_e - (\alpha_h + \alpha_j) - (d_h + d_j)\beta_i & \text{if } e = \{h, j\} \in E(J) \\ c_e - \alpha_j - d_j\beta_i & \text{if } e = \{i, j\} \in \delta(\{i\}). \end{cases} \tag{32}$$

Let us write a route $l \in \Omega_i$ like a sequence of edges in E, that is $l = (e_t)_{t=1}^{p}$ (in the case in which cycles are permitted, edges may appear more than once in the sequence). Thus, the reduced cost of such a route is given by the following expression:

$$\overline{c}_l = \sum_{t=1}^{p} \overline{c}_{e_t}. \tag{33}$$

It follows that in this case a column of minimum reduced cost can be computed as the solution of $|I'|$ shortest path problems with resource constraints. Moreover, the addition of any cut of the general form

$$\sum_{i \in I'} \tau_i z_i + \sum_{e \in E} \sum_{l \in \Omega} q_l^e \phi_e \lambda_l + \sum_{e \in \delta(I')} \varsigma_e y_e \leq \pi \tag{34}$$

produces a contribution to the computation of the reduced cost of the columns that can still be decomposed by edge, thus without breaking the shortest path structure of the pricing.

This is the case for all of the cuts valid for the two-index formulation of the CLRP after being translated to formulation (SPF) using identity (17).

4.3.2. Addition of the strong constraints and effect on the reduced costs

When a constraint cannot be written edge-by-edge, as for constraints (21), (23), (27), (28), (30) or (31), the contribution to the reduced cost cannot be decomposed edge by edge, and thus the original structure of the SPPRC is broken.

Indeed, consider a SRI for a clique C such that for $\chi(C) = \{i, j, k\}$ with dual variable $\sigma \leq 0$. The reduced cost $\bar{c}_l$ of a route $l \in \Omega$ that crosses *at least two* of those three customers must be augmented by $-\sigma$ units.

For the other strong constraints SDEG, y-SCC, SP-SFCI, SP-ESFCI or SFrCI, the contribution to the reduced cost is related to the simple intersection of path l with the sets describing the constraints. For instance, if we consider a SDEG constraint associated to a customer j and with dual variable $\sigma \geq 0$, then the reduced cost of a route l will be reduced by σ units *if l passes through node j*. Now, consider a constraint y-SCC for given sets $S \subseteq J$ and $S' \subset S$ as in (21) with dual value $\sigma \geq 0$. The contribution to the reduced cost will reduce it by σ units *if l intersects set S*. For a SP-SFCI or SP-ESFCI associated to sets $S \subset J$, $S' \subset S$, I' with dual variable $\sigma \geq 0$, the contribution to the reduced cost will reduce it by σ units *if route l crosses set S but is not linked to a facility in I'*. Finally, for the SFrCI associated to set S and partition $S_i, i = 1, \ldots, t$ and with dual variable $\sigma \geq 0$, the reduced cost must be reduced by σ units *once for each time that route l intersects either S or any of its subsets*.

4.3.3. The pricing problem

The pricing problem corresponds to solve $|I'|$ 2-cyc-SPPRC, one for each facility in I'. The resources associated to each label during the recursion are 1) vehicle load, 2) binary resources related to constraints SDEG, y-SCC, SP-SFCI, SP-ESFCI and SFrCI and 3) resources for taking into account the SRI. The algorithm used to solve these problems is based on dynamic programming (DP), as was done by several authors (Desrochers et al. 1992, Baldacci et al. 2008, Feillet et al. 2007, Jepsen et al. 2008, Righini and Salani 2008). Moreover, it is also possible to solve it by means of bidirectional DP (BDP). In classical uni-directional DP, paths are extended until reaching the depot node while ensuring that loads do not exceed capacity. In BDP, however, paths are extended until reaching half of the capacity for later joining paths pairwise. In this section we describe the 2-cyc-SPPRC algorithm used in the context of the CLRP. For general use of the dynamic programming method for solving the SPPRC we refer to the papers cited above. Let us denote by $V(L)$ the set of nodes served by the path represented by label L.

4.3.3.1 Resources description As said before, three different types of resources are considered in the problem: vehicle load resource; resources associated to constraints SDEG, y-SCC, SP-SFCI, SP-ESFCI and SFrCI; and resources associated to SRI.

Vehicle load The vehicle load is defined by an integer variable q that keeps track of the load of the current path. It is updated every time that a path is extended to a customer node.

Resources associated to SDEG, y-SCC, SP-SFCI, SP-ESFCI and SFrCI For each of the constraints SDEG, y-SCC, SP-SFCI and SP-ESFCI, the associated resource is defined by a single boolean variable that takes the value *true* if the path intersects the proper set as described before. We designate those sets as critical sets, and denote them by $S(C)$ for every constraint C. For each constraint SFrCI, there will be not one, but as many boolean variables as the size of the partition, plus one for the frame. Each of these variables will take the value *true* if the path crosses the proper set. Now, we do not have one but several critical sets that we denote by $S(C, k)$. Any time that one of these boolean variables passes from *false* to *true*, the reduced cost of the current path is reduced according to the value of the dual variable.

Resources associated to SRI For every clique C with $\chi(C) = \{i, j, k\}$ we associate three binary variables $r_C(k), k = 1, 2, 3$ that are initialized to 0 until the path crosses one of the customers, in which case the proper variable is set to 1, and the reduced cost of a path will be updated whenever $r_C(1) + r_C(2) + r_C(3)$ reaches the value 2.

4.3.3.2 The 2-cyc-SPPRC algorithm We first describe the definition of a label in the recursion of the dynamic programming algorithm. Then, we describe the dominance rules used to discard labels. After that, we describe a fathoming rule that can be aplied in order to also discard labels that cannot lead to a column of negative reduced cost. Next, we describe the path joining procedure to construct feasible paths from a given pair of labels. At the end, we describe the skeleton of the algorithm.

Label definition A label L is defined by

i. A node $v(L)$ which is the end node of the path represented by label L.

ii. A cost $\overline{c}(L)$ representing the reduced cost of the path represented by label L.

iii. A load resource $q(L)$ representing the load of the path represented by label L.

iv. Resources $res_C(L)$ associated to the binding constraints SDEG, y-SCC, SP-SFCI, SP-ESFCI, SFrCI and SRI. For constraints SFrCI and SRI we write $res_C(L, k)$ for the different sub-resources associated to these constraints.

v. An integer variable $v_{dom}(L)$ initially set to -1 and updated whenever L is found to be dominated by a label L', in which case we set $v_{dom}(L) = v(pred(L'))$.

vi. A boolean variable $proc(L)$ initialized to *false* and updated to *true* whenever the algorithm processes the label and inspects its neighbors.

vii. A pointer to the predecessor label $pred(L)$ of L.

viii. A list $succ(L)$ of pointers to the successors of L. $succ_i(L)$ denotes the i-th successor of label L.

Dominance rule Let L, L' be two labels. We denote $\text{SRI}_{LL'} = \{C \in \text{SRI} : \sum_k res_C(L,k) \leq 1 \text{ and } [\sum_k res_C(L',k) \geq 2 \text{ or } \exists\, k \text{ s.t. } res_C(L,k) < res_C(L',k)]\}$, $n_{C,L,L'} = |\{k : res_C(L,k) < res_C(L',k)\}|$ and $\text{OTH}_{L,L'} = \{C \in \text{SDEG} \cup y\text{-SCC} \cup \text{SP-SFCI} \cup \text{SP-ESFCI} : res_C(L) < res_C(L')\}$. We will say that L is dominated by L' if

i. $v(L) = v(L')$.

ii. $q(L) \geq q(L')$.

iii. $\overline{c}(L) \geq \overline{c}(L') - \sum_{C \in \text{SRI}_{LL'}} \sigma_C + \sum_{C \in \text{SFrCI}} n_{C,L,L'} \sigma_C + \sum_{C \in \text{OTH}_{LL'}} \sigma_C$.

The dominance rule is a direct application of the one used by Archetti et al. (2009) for the inclusion of SRI and k-path inequalities in the context of the VRP with split deliveries and time windows (VRPSDTW). A label L that is dominated by another label L' cannot be directly eliminated unless $v(pred(L)) = v(pred(L'))$ or if $v_{dom}(L) \notin \{-1, v(pred(L'))\}$. In that case, label L is removed and recursively we also remove all of its successors in $succ(L)$. Otherwise, $v_{dom}(L)$ is set to $v(pred(L'))$. Note that the inclusion of SDEG constraints allows to weaken the dominance rule with respect to a traditional elementarity constraint, in which the condition for dominance would be $res_C(L) \geq res_C(L')$ for each $C \in \text{SDEG}$.

Fathoming rule In addition to the dominance criterion, a fathoming rule can be applied if a lower bound on the cost of extending a path can be computed. Formally, let L be a label and let $LB(L)$ be a lower bound on the reduced cost that can be obtained by extending L, computed as follows. First of all, discard SRI as their dual variables are negative. For every binding strong constraint $C \in \mathcal{C} = \text{SDEG} \cup y\text{-SCC} \cup \text{SP-SFCI} \cup \text{SP-ESFCI} \cup \text{SFrCI}$, with dual variables $(\sigma_C)_{C \in \mathcal{C}}$, and for every edge e crossing the critical sets related to these constraints, we decrease the reduced cost of that edge by $\sigma_C/2$ units. We refer to this procedure as *under-estimation of constraint* C. As a route that crosses a customer set S must have at least two edges in $\delta(S)$ then the reduced cost of a path computed in this way will in fact be a lower bound on the real reduced cost. We then solve the related 2-cyc-SPPRC with no resources associated to strong constraints, and compute functions f, g and π as follows:

$$f(p,i) = \min\{\overline{c}(L) : v(L) = i, q(L) \leq Q - p + d_i\} \tag{35}$$

$$\pi(p,i) = v(pred(arg\,min\{f(p,i)\})) \tag{36}$$

$$g(p,i) = \min\{\overline{c}(L) : v(L) = i, q(L) \leq Q - p + d_i, v(pred(L)) \neq \pi(p,i)\} \tag{37}$$

For a constraint $C \in \text{SFrCI}$ and a customer $i \in J$, let $n_{C,i} = |\{k : i \in S(C,k)\}|$. Also, let

$$h(L) = \begin{cases} f(q(L), v(L)) & \text{if } \pi(q(L), v(L)) \neq v(pred(L)) \\ g(q(L), v(L)) & \text{otherwise.} \end{cases}$$

.

A lower bound on the reduced cost reachable by extending label L can be computed as

$$LB(L) = \bar{c}(L) + h(L) + \frac{1}{2} \sum_{\substack{C \in \mathcal{C} \setminus \text{SFrCI} \\ i \in S(C)}} \sigma_C + \frac{1}{2} \sum_{C \in \text{SFrCI}} n_{C,i} \sigma_C. \tag{38}$$

The two sums aim to compensate the fact that the contribution of the under-estimated constraints $C \in \mathcal{C}$ is being considered at least 1.5 times in $\bar{c}(L)$ and $h(L)$ whenever $i \in S(C)$ or $n_{C,i} > 0$, thus tightening $LB(L)$. If a label L is such that $LB(L) > 0$, then L can be discarded. Similar fathoming rules have been implemented by Christofides et al. (1981), Baldacci et al. (2008, 2010a) and Baldacci et al. (2010b), for instance. Note that we have used unidirectional DP for computing functions f, g, π. From an implementation point of view, it only differs from the BDP in the fact that now all labels are inspected for extension and not only those whose load is less or equal than $Q/2$, so at the end the joining of paths is not necessary. Note also that this fathoming procedure can be generalized (and also strengthened) by keeping as resources, thus without under-estimating, the k constraints $C \in \mathcal{C}$ with the largest duals, where k is a parameter defined *a priori*. After doing a series of experiments, we let $k = \min\{20, |\mathcal{C}|/5\}$. For these constraints, the coefficients in the sums in (38) can now be lifted to 1, as the contribution to the reduced cost of a customer such that $i \in S(C)$ or $n_{C,i} > 0$ is being counted twice.

Path joining As the labeling algorithm is bidirectional, the labels must be joined to construct feasible paths. Given two labels L, L' such that $v(L) = v(L')$ and $q(L) + q(L') \leq Q + d_{v(L)}$, they will produce a feasible path (one that satisfies capacity constraints and such that its reduced cost is negative) if

i. $\min\{q(L), q(L')\} \geq \frac{q(L)+q(L')-d_{v(L)}}{2}$

ii. $\max\{q(L), q(L')\} \leq \frac{q(L)+q(L')+d_{v(L)}}{2}$

iii. $v(pred(L)) < v(pred(L'))$

iv. the reduced cost of the concatenated path $P = (L, L')$ is negative.

The first two conditions are the median conditions (Baldacci et al. 2008) that ensure that labels L and L' are the closest possible to half of the load. The third condition ensures that if path $P = (L, L')$ is kept, then path $P' = (L', L)$ will be discarded. This way, symmetric or repeated paths will not be added to the master problem.

The dynamic programming algorithm Let us describe the labeling algorithm by means of a pseudo-code. Let L_0 be the label representing an empty path starting at the facility, such that all of the resources are set at their default values. Also, let us note that labels will be stored in buckets, and let $B(q, v)$ be the bucket storing labels L whose loads are $q(L) = q$ and such that $v(L) = v$.

Algorithm 1 2-cyc-SPPRC

Compute functions f, g, π using DP.
$B(0,0) \leftarrow \{L_0\}, \mathcal{V} \leftarrow \{0\}, \mathcal{R} \leftarrow \emptyset$.
repeat
  Take node v from $\mathcal{V}$ and set $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v\}$.
  for $q = 0$ **to** $Q/2$ **do**
    for all $L \in B(q, v)$ such that $proc(L) =$ **false do**
      Set $proc(L) \leftarrow$ **true**.
      for all $w \in$ Neighbors of v, $w \neq 0$ **and** $q(L) + d_w \leq Q$ **and** $pred(L) \neq w$ **do**
        Create L' such that $v(L') = w$ and $pred(L') = L$. Update resources accordingly.
        Apply fathoming rule and eventually discard L'.
        Apply dominance rule and eventually discard L'.
        if L' has not been discarded **then**
          Make $B(q(L'), w) \leftarrow B(q(L'), w) \cup \{L'\}$.
          Apply dominance rule and eventually delete other labels in $B(q(L'), w)$.
          Make $\mathcal{V} \leftarrow \mathcal{V} \cup \{w\}$.
        end if
      end for
    end for
  end for
until $V = \emptyset$
Join paths $\{(L, L') : v(L) = v(L') = v, q(L) + q(L') \leq Q + d_v\}$ and fill $\mathcal{R}$
return $\mathcal{R}$

4.3.4. Computing lower bounds

When pricing problems are solved to optimality, it is possible to obtain a lower bound on the problem. This lower bound can then be used for fathoming the current node as well as for early termination criteria. The following proposition provides a way of computing a lower bound on the CLRP.

Proposition 4.1. *Let $\bar{c}_{\min}$ be the minimum reduced cost at the current iteration for columns in Ω, and let $\bar{z}$ be the value of the master problem at the current iteration. Also, let $K_{\max}$ be an upper bound on the number of vehicles that serve two or more customers. A valid lower bound for the CLRP is given by*

$$z_{LB} = \bar{z} + K_{\max}\bar{c}_{\min}. \tag{39}$$

Proof Let σ be the dual variables of the linear relaxation of problem (SPF). Let $(\bar{c}_l)_{l\in\Omega}$ be the reduced costs of columns serving two or more customers, that depend on the duals σ. The Lagrangean dual of this problem, that can be written in the following form, provides a valid lower bound for the CLRP

$$L(\sigma) = \bar{z} + \min\{\sum_{l\in\Omega} \bar{c}_l\lambda_l : \sum_{l\in\Omega} \lambda_l \leq K_{\max}\}. \tag{40}$$

But now, as $\bar{c}_{\min} \leq 0$ then $\min\{\sum_{l\in\Omega} \bar{c}_l\lambda_l : \sum_{l\in\Omega} \lambda_l \leq K_{\max}\} \leq K_{\max}\bar{c}_{\min}$. □

For every candidate set I' we use $K_{\max} = K_m(I')$ as described in the first bounding procedure.

4.4. Enumeration of remaining columns

For each subset of facilities I' as obtained after the first bounding procedure and not discarded after the second procedure, let z_{LB} and σ be the lower bound at the end of the second bounding procedure and the dual variables associated to such lower bound. If procedure ENUM-ESPPRC was successful to generate the column set $\mathcal{P}$, we simply compute the reduced cost of columns in $\mathcal{P}$ and add to the master problem those columns l such that $\overline{c}_l < z_{UB} - z_{LB}$. We then solve the resulting integer problem using a general-purpose solver such as CPLEX. If, however, we were not able to obtain set $\mathcal{P}$, we first check whether the upper bound z_{UB} improved during the second bounding procedure after the consideration of set I'. In this case, we run again algorithm ENUM-ESPPRC but now with the updated upper bound, as the performance of algorithm ENUM-ESPPRC depends strongly on the gap $z_{UB} - z_{LB}$. Otherwise, we start the following procedure with the hope of getting a better upper bound (if any), and in the worst case it gives us a method for tightening the gap.

i. Let $\Delta \leftarrow (z_{UB} - z_{LB})/10$. Set $k \leftarrow 1$.

ii. Let $z'_{UB} \leftarrow z_{LB} + k\Delta$ and try to generate all of the columns whose reduced costs are smaller or equal than $k\Delta$. If more than $\Delta_{max} = 10^6$ columns are found or if we run out of memory, we exit. Otherwise we go to step (iii).

iii. Solve the resulting integer problem to optimality. If a new upper bound was found with value $z^* < z_{UB}$, set $z_{UB} \leftarrow z^*$ and $z_{LB} \leftarrow \min\{z_{UB}, z'_{UB}\}$. If $z_{LB} = z'_{UB}$ then exit. Otherwise, if either $z'_{UB} < z^*$ or the problem was solved to optimality but no integer solution was found with value less than z_{UB}, set $z_{LB} = z'_{UB}$. If $k < 10$ do $k \leftarrow k + 1$ and go back to (ii).

This method generalizes the one proposed by Baldacci et al. (2010b) by artificially lowering the optimality gap and iteratively increasing it, thus reducing the negative impact of an initial upper bound of poor quality. Let us describe the algorithm for solving the column enumeration problem. This algorithm is a variation of the Elementary SPPRC (ESPPRC) and we call it ENUM-ESPPRC.

4.4.1. The column enumeration algorithm

Algorithm ENUM-ESPPRC is based on the solution of the ESPPRC, and so as the 2-cyc-SPPRC, is solved by means of bidirectional dynamic programming. The method presented in this paper differs from the one proposed by Baldacci et al. (2010b) mainly in the fathoming rule that considers the inclusion of the strong constraints in the value of the completion bound for a given path label. As for the description of the 2-cyc-SPPRC, we first describe the definition of a label in the recursion of the dynamic programming algorithm. Then, we describe the dominance rules used to discard labels. After that, we describe a fathoming rule that can be applied in order to also discard labels that cannot lead to a column of reduced cost smaller than a desired threshold. Next, we describe the path joining procedure to build feasible paths from a given pair of labels. At the end, we describe the skeleton of the algorithm.

Label definition We define a label L containing the same information as for the 2-cyc-SPPRC algorithm plus

i. A cost $c(L)$ representing the cost of the path represented by label L.

ii. Additional resources associated to nodes. For every customer $j \in J$ we associate a boolean variable $res_j(L)$ equal to $true$ if $j \in V(L)$, 0 otherwise.

Dominance rule Given two labels L, L', we say that L is dominated by L' if

i. $v(L) = v(L')$

ii. $V(L) = V(L')$

iii. $c(L) \geq c(L')$

Now, dominance is done with respect to the costs instead of the reduced costs. A Label L that is found to be dominated by another label L' is removed, and recursively also all of its successors.

Fathoming rule A similar fathoming rule as the one used for the 2-cyc-SPPRC can be applied. Indeed, it only differs from the one used for the 2-cyc-SPPRC in the parameter k for the number of non under-estimated constraints that is set to $k = |\mathcal{C}|$. Thus, a lower bound $LB(L)$ on the reduced cost of a label L after extending it is given by

$$LB(L) = \overline{c}(L) + h(L) + \sum_{\substack{C \in \mathcal{C} \setminus \text{SFrCI} \\ i \in S(C)}} \sigma_C + \sum_{C \in \text{SFrCI}} n_{C,i} \sigma_C, \tag{41}$$

where $h(L)$, σ_C and $n_{C,i}$ are as defined for the fathoming rule of the 2-cyc-SPPRC. Now, a label L will be discarded if $LB(L) \geq z_{UB} - z_{LB}$.

Path joining A similar joining procedure can be applied to algorithm ENUM-ESPPRC as with the 2-cyc-SPPRC, with the main difference that now cycles are not allowed at all. Given two labels L, L' such that $v(L) = v(L')$ and $q(L) + q(L') \leq Q + d_{v(L)}$, they will produce a feasible path (one that satisfies capacity constraints and such that its reduced cost is smaller than the desired threshold) if

i. $\min\{q(L), q(L')\} \geq \frac{q(L)+q(L')-d_{v(L)}}{2}$

ii. $\max\{q(L), q(L')\} \leq \frac{q(L)+q(L')+d_{v(L)}}{2}$

iii. $v(pred(L)) < v(pred(L'))$

iv. $V(L) \cap V(L') = \{0, v(L)\}$

v. the reduced cost of the concatenated path $P = (L, L')$ is smaller than $z_{UB} - z_{LB}$.

Now, condition (iv) ensures that paths L, L' only share the facility and the joining node.

The dynamic programming algorithm Let us describe the labeling algorithm by means of a pseudo-code. Just as before, label L_0 represents an empty path starting at the facility, such that all of the resources are set at their default values. Labels will be stored in buckets, and let $B(q, v)$ be the bucket storing labels L whose loads are $q(L) = q$ and such that $v(L) = v$.

Algorithm 2 ENUM-ESPPRC

Compute functions f, g, π using DP.
$B(0,0) \leftarrow \{L_0\}, \mathcal{V} \leftarrow \{0\}, \mathcal{R} \leftarrow \emptyset$
repeat
  Take node v from $\mathcal{V}$ and set $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v\}$
  for $q = 0$ **to** $Q/2$ **do**
    for all $L \in B(q, v)$ such that $proc(L) =$ **false do**
      Set $proc(L) \leftarrow$ **true**.
      for all $w \in$ Neighbors of v, $w \neq 0$ **and** $q(L) + d_w \leq Q$ **and** $w \notin V(L)$ **do**
        Create L' such that $v(L') = w$ and $pred(L') = L$. Update resources accordingly.
        Apply fathoming rule and eventually discard L'.
        Apply dominance rule and eventually discard L'.
        if L' has not been discarded **then**
          Make $B(q(L'), w) \leftarrow B(q(L'), w) \cup \{L'\}$.
          Apply dominance rule and eventually delete other labels in $B(q(L'), w)$.
          Make $\mathcal{V} \leftarrow \mathcal{V} \cup \{w\}$.
        end if
      end for
    end for
  end for
until $V = \emptyset$
Join paths $\{(L, L') : v(L) = v(L') = v, q(L) + q(L') \leq Q + d_v\}$ and fill $\mathcal{R}$.
return $\mathcal{R}$

4.5. Computational issues

We now make some observations that can help to accelerate the algorithm.

4.5.1. Initial set of columns

An initial set of columns is required in column generation algorithms. Indeed, at every iteration of the CG, a feasible solution of the master problem is needed for running the pricing algorithms. In our algorithm, we let the initial set of columns contain only the single-customer variables y. Additionally, we also add slack and artificial variables to the formulation so the problem will always have a feasible solution.

4.5.2. Stabilization of the column generation

With the aim of reducing the oscillation of the dual variables during the first iterations of the column generation process, we use a box-pen method (du Merle et al. 1999) for stabilizing the duals of the degree constraints (12). For every set $I' \in \mathcal{I}$, the centers are initially set to the optimal dual variables of the degree constraints (1) after performing the first bounding procedure.

4.5.3. Column pool management

For some instances, the quantity of columns added can be huge and, moreover, most of them will be useless. In fact, it is known that at the beginning of the column generation process, many columns are generated that soon will become non-basic for the rest of the algorithm. We keep a pool of columns and keep track of the number of consecutive iterations that columns have been non-basic. Every 30 iterations we check and delete all columns having been inactive for more than 30 iterations. Note that after the creation of set $\mathcal{P}$ during the second bounding procedure, the columns deleted from the problem must be inserted back into $\mathcal{P}$.

4.5.4. Memory management

The dynamic programming algorithms can be very demanding in terms of memory. In fact, every new created label needs to be allocated in memory. In this context, the *new* and *delete* operators of C++ (or *malloc* and *free* operators in the case of C) can be very inefficient. We have decided to manage our own memory pool, in which dynamic memory is allocated in chunks of 400 MB. The newly created labels are thus allocated inside the previously allocated memory.

5. Computational Experience

We have run our method on an Intel Xeon E5462, 3.0 Ghz processor with 16GB of memory. The code was compiled with the Intel C++ compiler v11.0 and executed on Linux, kernel 2.6. Linear and integer programs were solved by CPLEX 12.2. The pricing algorithms 2-cyc-SPPRC and ENUM-ESPPRC have been coded in C++ using the same compiler as before. The algorithm has been tested over five sets of instances from the literature, containing in total 71 instances. The first family ($\mathcal{F}_1$) has been adapted by Barreto (2004) from other vehicle routing problems in the literature and contains 16 instances with capacitated vehicles and facilities. The second set of instances ($\mathcal{F}_2$) has been developed by Belenguer et al. (2010) and contains 30 instances with capacitated vehicles and facilities. The third set of instances ($\mathcal{F}_3$) has been introduced by Akca et al. (2009) and contains 12 instances with capacitated vehicles and facilities. The fourth set of instances ($\mathcal{F}_4$) has been introduced by Tuzun and Burke (1999) and contains 9 instances with capacitated vehicles and uncapacitated facilities. The fifth and last set of instances ($\mathcal{F}_5$) has been introduced by Baldacci et al. (2010b) and contains 4 instances with capacitated vehicles and uncapacitated facilities. The dimensions of the instances vary from very small instances with 12 customers and 2 facilities up to very large instances with 199 customers and 14 facilities. We compare our results against those obtained by other exact algorithms, namely the methods of Belenguer et al. (2010), Contardo et al. (2010) and Baldacci et al. (2010b). We use as upper bound the best solution available in the literature for every instance. In Tables 1-5 we present the detailed results obtained by our algorithm for every instance and for each of the three bounding procedures. The columns in these tables are as follows:

i. *Instance*: name of the instance.

ii. z_{UB}: objective function value of the best feasible solution available in the literature.

iii. z^*: objective function value of the best feasible solution found by our algorithm. The text in bold characters indicates that this value is strictly lower than the one in column labeled z_{UB}.

iv. gap_1, t_1: gap obtained and CPU time taken by the first bounding procedure. The gap is computed as follows: $(z^* - z_{LB_1})/z^* \times 100$.

v. gap_2, t_2: gap obtained and CPU time taken by the second bounding procedure.

vi. gap_3, t_3: gap obtained and CPU time taken by the column enumeration procedure.

vii. $|\mathcal{I}|$: number of subsets obtained by the first bounding procedure.

viii. $|\mathcal{R}_{1,2}|$: maximum number of columns found by the procedure ENUM-ESPPRC after the second bounding procedure and the final enumeration step, respectively. This maximum is taken over all subsets $I' \subseteq \mathcal{I}$.

ix. t: overall CPU time.

As shown in these tables, our algorithm is capable of solving 58 out of the 71 instances considered. Moreover, all instances of families $\mathcal{F}_1$ and $\mathcal{F}_3$ (28 in total) are solved to optimality, and for none of them was procedure ENUM-ESPPRC called during the third bounding procedure. Finally, instances Chr-75x10ba, ppw-50x5-2b, ppw-100x5-2b and ppw-200x10-3a were solved to optimality for the first time, and we have improved the best feasible solution for three more instances (ppw-100x5-0b, P113112 and P131112). As a matter of fact, our method is able to solve all instances with 85 customers or less.

We first compare our method against the branch-and-cut algorithms of Belenguer et al. (2010) and of Contardo et al. (2010). In Tables 6-8 we establish the gaps and CPU times obtained by every algorithm on three sets of instances. In these tables, headers BBPPW, CCG-BC and CCG-BCP stand for the methods of Belenguer et al. (2010), Contardo et al. (2010) and this work, respectively. In the case of method CCG-BC we consider the branch-and-cut algorithm with the two-index vehicle-flow formulation of the problem. In the case of the branch-and-cut algorithms, columns labeled gap_{lr}, t_{lr}, gap and t stand for the gaps and CPU times for the root node relaxation and after the whole branching tree (with a maximum CPU time of 2 hours). In the case of method CCG-BCP, columns gap_1, t_1 stand for the gap obtained after the first bounding procedure, and columns labeled gap, t stand for the final gap and the total CPU time spent by the method. We highlight in bold characters whenever a method dominates the other two in terms of bound quality. First of all, the first bounding procedure produces better bounds than the flow-based algorithms at the root node. This is not surprising since this procedure uses the code of CCG-BC for doing a partial branch-and-bound on the location variables. At the end, our method is able to produce tighter gaps than the other two. Although a CPU-based comparison can be difficult (because each algorithm was run on different machines), it is worth noting that our method was some orders of magnitude faster on the instances of family $\mathcal{F}_3$ (Table 8). Moreover, we can solve 48 of the considered instances, 20 more than BBPPW and 18 more than CCG-BC.

Finally, we compare the proposed methodology against the column generation method of Baldacci et al. (2010b). In Tables 9-12 we compare the three bounding procedures introduced in this paper against the similar bounds used in the method of Baldacci et al. (2010b). Note that, although the second and third bounding procedures in both methods are very similar, the first bounding used by Baldacci et al. (2010b) is a relaxation of the set-partitioning formulation, while in our case it is based on the two-index vehicle-flow formulation of the problem. In these tables the legend is analogous to that used for the previous set of tables. We also highlight in bold characters whenever a bound dominates the other. As shown in these tables, our method is able to produce tighter bounds than that of Baldacci et al. (2010b) for most instances and for every bounding procedure. Our first bounding procedure is quite effective whenever branching decisions on the location variables have a significant impact on either the bounds or the feasibility of the problem. Indeed, this is the case for all sets of instances except for $\mathcal{F}_5$. Our first bounding procedure obtains smaller gaps than that of Baldacci et al. (2010b) in 51 out of the 71 instances considered. For the second bounding procedure, our algorithm obtains smaller gaps in 43 out of the 71 instances. This shows the strength of the set-partitioning formulation with the additional cuts. Our third bounding procedure, although it can be very time consuming, is shown to be effective for solving instances Chr-75x10ba and ppw-200x10-3a in which the initial upper bounds are significantly improved during this procedure. In general, our algorithm is able to solve four instances that are not solved by the method of Baldacci et al. (2010b), and improves the best known feasible solution in three other instances. However, for the instances in family $\mathcal{F}_5$ our method is outperformed by that of Baldacci et al. (2010b). The overall results suggest that our method is competitive against the one of Baldacci et al. (2010b). This is the result of several refinements with respect to their method, namely the use of the new cuts, as well as the use of efficient pricing algorithms that properly handle these new cuts. This includes the use of stronger fathoming procedures based on the solution of a 2-cyc-SPPRC with resources.

6. Concluding remarks

In this paper, we have presented an exact method for solving the CLRP. The methodology consists in formulating the CLRP as a set-partitioning problem that is solved in three stages: in a first stage we consider the two-index formulation and branch on the location variables. This strategy works well for instances in which branching decisions on the location variables have a significant impact on the feasibility or the bound at the resulting nodes in the branching tree. The remaining gap is then closed by sequentially applying two procedures, both based on the set-partitioning formulation and solved by means of column-and-cut generation. The algorithm proposed in this paper is able to produce the tightest gaps on a large number of instances. In addition, it has solved to optimality four previously open instances and improved the best known feasible solution for three additional ones. The methodology can be easily adapted to solve other routing problems. For instance, it would be interesting to measure the impact of y-SCC and SFrCI cuts on solving hard instances of the CVRP. With respect to the pricing algorithm introduced in this paper, the consideration of SDEG cuts allows to get lower bounds that are comparable to those obtained when pricing on elementary routes in a fraction of the computational effort. Indeed, in most cases only a fraction of SDEG cuts need to be added to the master problem to obtain significant improvements in the lower bound. Moreover, we show how to take advantage of this pricing problem in the computation of tight fathoming rules that speed up the whole algorithm. Further research related to the methodology introduced in this paper should address the development of new cutting planes for the set-partitioning formulation and to adapt some of them to other routing problems.

Acknowledgements

The authors would like to thank Simon Spoorendonk and Enrico Bartolini for their insightful comments. Thanks are also due to the *Natural Sciences and Engineering Research Council of Canada* (NSERC) and *Le fonds québécois de la recherche sur la nature et les technologies* (FQRNT) for their financial support.

Instance	z_{UB}	z^*	gap$_1$	$\lvert\mathcal{I}\rvert$	t_1	gap$_2$	$\lvert\mathcal{R}_1\rvert$	t_2	gap$_3$	$\lvert\mathcal{R}_2\rvert$	t_3	t
Perl83-12x2	204.00	204.00*	0.00	1	0.02	0.00	4	0.01	0.00	0	0.00	0.03
Gas67-21x5	424.90	424.90*	1.61	2	0.25	0.00	17	0.08	0.00	0	0.00	0.33
Gas67-22x5	585.11	585.11*	0.10	1	0.05	0.00	24	0.18	0.00	0	0.00	0.23
Min92-27x5	3062.02	3062.02*	0.00	1	0.21	0.00	15	0.10	0.00	0	0.00	0.31
Gas67-29x5	512.10	512.10*	1.88	1	0.44	0.00	2077	3.76	0.00	0	0.00	4.20
Gas67-32x5	562.22	562.22*	1.24	1	0.57	0.00	12512	5.95	0.00	0	0.00	6.52
Gas67-32x5-2	504.33	504.33*	0.01	1	0.51	0.00	9	0.19	0.00	0	0.00	0.70
Gas67-36x5	460.37	460.37*	0.00	0	1.04	0.00	0	0.00	0.00	0	0.00	1.04
Chr69-50x5ba[1]	565.62	565.62*	1.58	2	6.53	0.00	10797	5.55	0.00	0	0.00	12.08
Chr69-50x5be[2]	565.60	565.60*	2.14	5	8.70	0.00	11928	18.66	0.00	0	0.00	27.36
Perl83-55x15	1112.06	1112.06	1.96	200	197.21	0.00	26860	43.54	0.00	0	0.00	240.75
Chr69-75x10ba[1]	886.30	**844.40***	7.27	489	1243.61	0.48	2199569	5749.77	0.00	0	13.04	7006.42
Chr69-75x10be[2]	848.85	848.85*	6.71	195	485.76	0.00	442577	3945.23	0.00	0	0.00	4430.99
Chr69-75x10bmw[3]	802.08	802.08*	6.02	117	207.56	0.00	1172461	1733.37	0.00	0	0.00	1940.93
Perl83-85x7	1622.50	1622.50*	1.65	19	76.12	0.00	767712	101.80	0.00	0	0.00	177.92
Chr69-100x10	833.43	833.43*	1.81	27	419.35	0.00	771623	1130.42	0.00	0	0.00	1549.77
Average			2.12		165.50	0.03		796.16	0.00		0.81	962.47

[1] Instance used by Barreto (2004).
[2] Instance used by Belenguer et al. (2010).
[3] Instance used by Baldacci et al. (2010b).
* Optimal solution.

Table 1: Results on family $\mathcal{F}_1$

Instance	z_{UB}	z^*	gap$_1$	$\lvert\mathcal{I}\rvert$	t_1	gap$_2$	$\lvert\mathcal{R}_1\rvert$	t_2	gap$_3$	$\lvert\mathcal{R}_2\rvert$	t_3	t
ppw-20x5-0a	54793	54793*	3.06	3	0.29	0.00	372	0.31	0.00	0	0.00	0.60
ppw-20x5-0b	39104	39104*	0.00	0	0.03	0.00	0	0.00	0.00	0	0.00	0.03
ppw-20x5-2a	48908	48908*	2.40	2	0.14	0.00	587	0.42	0.00	0	0.00	0.56
ppw-20x5-2b	37542	37542*	0.00	0	0.02	0.00	0	0.00	0.00	0	0.00	0.02
ppw-50x5-0a	90111	90111*	5.95	3	7.25	0.15	51513	9.69	0.00	0	0.14	17.08
ppw-50x5-0b	63242	63242*	3.55	1	4.18	0.58	2456549	296.71	0.00	0	177.14	478.03
ppw-50x5-2a	88298	88298*	3.92	3	6.31	0.00	21673	7.18	0.00	0	0.00	13.49
ppw-50x5-2b	67340	**67308***	3.77	3	2.84	0.00	346704	441.49	0.00	0	0.00	444.33
ppw-50x5-2a'	84055	84055*	1.99	2	7.44	0.03	134118	22.01	0.00	0	0.08	29.53
ppw-50x5-2b'	51822	51822*	0.69	2	1.49	0.00	40913	5.76	0.00	0	0.00	7.25
ppw-50x5-3a	86203	86203*	3.93	3	13.19	0.82	113809	25.12	0.00	0	55.93	94.24
ppw-50x5-3b	61830	61830*	2.30	4	5.78	0.00	937460	80.95	0.00	0	0.00	86.73
ppw-100x5-0a	274814	274814*	4.55	4	144.98	0.20	469031	238.71	0.00	0	62.44	446.13
ppw-100x5-0b	214392	**213568**	3.14	2	119.05	0.46	Δ_{max}	24902.90	0.29	248557	22824.30	47846.20
ppw-100x5-2a	193671	193671*	3.52	1	36.06	0.06	64646	19.92	0.00	0	0.55	56.53
ppw-100x5-2b	157173	**157095***	2.14	2	42.68	0.10	978507	10486.60	0.00	0	2047.04	12576.30
ppw-100x5-3a	200079	200079*	3.55	1	35.56	0.19	529311	53.88	0.00	0	34.30	123.74
ppw-100x5-3b	152441	152441*	1.95	1	23.85	0.00	210300	446.04	0.00	0	0.00	469.89
ppw-100x10-0a	289017	289017	5.77	5	1147.03	1.70	Δ_{max}	1623.27	1.27	251056	43573.30	46343.60
ppw-100x10-0b	234641	234641	4.44	5	163.75	2.06	Δ_{max}	15156.30	1.94	321769	22772.60	38092.70
ppw-100x10-2a	243590	243590*	2.82	8	169.38	0.32	1811787	729.95	0.00	0	1353.58	2252.91
ppw-100x10-2b	203988	203988*	0.79	3	72.55	0.00	161828	252.17	0.00	0	0.00	324.72
ppw-100x10-3a	252421	252421	6.02	18	1425.99	2.00	Δ_{max}	1026.68	1.60	166861	22460.70	24913.40
ppw-100x10-3b	204597	204597	4.02	6	150.36	1.49	Δ_{max}	50030.40	1.36	151231	30015.40	80196.10
ppw-200x10-0a	479425	479425	8.66	31	3861.00	1.28	Δ_{max}	13040.60	1.15	894085	32897.40	49799.00
ppw-200x10-0b	378773	378773	5.32	10	3367.07	1.19	Δ_{max}	148669.00	1.19	Δ_{max}	85063.50	237100.00
ppw-200x10-2a	450468	450468	5.06	3	359.83	0.88	Δ_{max}	8838.32	0.80	683782	22043.00	31241.20
ppw-200x10-2b	374435	374435	3.10	3	566.81	0.42	Δ_{max}	61771.20	0.37	Δ_{max}	30157.70	92495.60
ppw-200x10-3a	472898	**469433***	6.61	16	3788.52	0.12	Δ_{max}	10313.30	0.00	169099	4636.90	18738.80
ppw-200x10-3b	364178	364178	4.92	6	2482.11	1.00	Δ_{max}	37910.00	1.00	Δ_{max}	4621.82	45013.90
Average			3.60		600.18	0.50		12879.96	0.37		10826.59	24306.75
Average on solved instances			2.87		218.13	0.13		1171.51	0.00		418.41	1808.05
Average on unsolved instances			5.04		1364.30	1.25		36296.87	1.10		31642.97	69304.17

* Optimal solution.

Table 2: Results on family $\mathcal{F}_2$

Instance	z_{UB}	z^*	gap$_1$	$\lvert\mathcal{I}\rvert$	t_1	gap$_2$	$\lvert\mathcal{R}_1\rvert$	t_2	gap$_3$	$\lvert\mathcal{R}_2\rvert$	t_3	t
cr30x5a-1	819.52	819.52*	2.89	2	0.60	0.00	2891	1.85	0.00	0	0.00	2.45
cr30x5a-2	821.50	821.50*	3.73	1	0.38	0.00	857	3.34	0.00	0	0.00	3.72
cr30x5a-3	702.30	702.30*	0.00	1	0.44	0.00	14	0.06	0.00	0	0.00	0.50
cr30x5b-1	880.02	880.02*	2.69	2	1.01	0.00	2963	3.56	0.00	0	0.00	4.57
cr30x5b-2	825.32	825.32*	1.22	1	0.97	0.00	29	0.27	0.00	0	0.00	1.24
cr30x5b-3	884.60	884.60*	2.33	1	0.92	0.00	31	0.31	0.00	0	0.00	1.23
cr40x5a-1	928.10	928.10*	3.30	7	3.99	0.00	14135	10.68	0.00	0	0.00	14.67
cr40x5a-2	888.42	888.42*	2.80	3	2.25	0.19	53615	9.58	0.00	0	0.05	11.88
cr40x5a-3	947.30	947.30*	3.02	4	3.09	0.00	9166	8.27	0.00	0	0.00	11.36
cr40x5b-1	1052.04	1052.04*	5.78	8	5.49	0.00	2522	5.00	0.00	0	0.00	10.49
cr40x5b-2	981.54	981.54*	2.09	3	1.63	0.00	329	2.14	0.00	0	0.00	3.77
cr40x5b-3	964.33	964.33*	2.00	1	1.67	0.00	33	1.01	0.00	0	0.00	2.68
Average			2.65		1.87	0.02		3.84	0.00		0.00	5.71

* Optimal solution.

Table 3: Results on family $\mathcal{F}_3$

Instance	z_{UB}	z^*	gap$_1$	$\lvert\mathcal{I}\rvert$	t_1	gap$_2$	$\lvert\mathcal{R}_1\rvert$	t_2	gap$_3$	$\lvert\mathcal{R}_2\rvert$	t_3	t
P111112	1467.68	1467.68*	4.63	42	885.82	0.00	2065379	3581.25	0.00	0	0.00	4467.07
P111212	1394.80	1394.80*	4.67	52	859.13	0.00	3429771	20689.10	0.00	0	0.00	21548.30
P112112	1167.16	1167.16*	2.74	1	75.47	0.00	35633	235.30	0.00	0	0.00	310.77
P112212	791.66	791.66*	1.36	1	21.32	0.00	4800861	1818.15	0.00	0	0.00	1839.47
P113112	1245.45	**1238.24**	3.95	13	454.58	0.61	Δ_{max}	118970.00	0.34	Δ_{max}	52278.40	171703.00
P113212	902.26	902.26*	0.49	2	51.30	0.00	8502	101.93	0.00	0	0.00	153.23
P131112	1900.70	**1896.98**	6.24	206	11926.90	0.85	Δ_{max}	299823.00	0.64	Δ_{max}	38047.20	349797.00
P131212	1965.12	1965.12	8.43	282	10682.50	0.95	Δ_{max}	207942.00	0.82	258029	33676.30	252301.00
P132112	1443.33	1443.32*	5.68	16	3582.12	0.00	1120735	15021.90	0.00	0	0.00	18604.00
Average			4.24		3171.02	0.27		74242.51	0.20		13777.99	91191.54
Average on solved instances			3.26		912.53	0.00		6907.94	0.00		0.00	7820.47
Average on unsolved instances			6.21		7687.99	0.80		208911.67	0.60		41333.97	257933.67

* Optimal solution.

Table 4: Results on family $\mathcal{F}_4$

Instance	z_{UB}	z^*	gap$_1$	$\lvert\mathcal{I}\rvert$	t_1	gap$_2$	$\lvert\mathcal{R}_1\rvert$	t_2	gap$_3$	$\lvert\mathcal{R}_2\rvert$	t_3	t
M-n150x14a	1352.93	1352.93*	9.61	2331	34342.60	0.09	3251152	351540.00	0.00	0	1.32	385884.00
M-n150x14b	1212.46	1212.46*	7.50	2465	39786.80	0.20	4507242	352454.00	0.00	0	30.68	392272.00
M-n199x14a	1644.35	1644.35*	12.06	1598	64412.40	0.15	4557968	616522.00	0.00	0	10.81	680945.00
M-n199x14b	1480.43	1480.43*	10.22	1513	67992.50	0.09	2141465	1006620.00	0.00	0	1.98	1074610.00
Average			9.85		51633.57	0.13		581784.00	0.00		11.20	633427.75

* Optimal solution.

Table 5: Results on family $\mathcal{F}_5$

Instance	z_{UB}	z^*	BBPPW				CCG-BC				CCG-BCP			
			gap_{lr}	t_{lr}	gap	t	gap_{lr}	t_{lr}	gap	t	gap_1	t_1	gap	t
Perl83-12x2	204.00	203.98					0.61	0.01	0.00	0.02	**0.00**	0.02	0.00	0.03
Gas67-21x5	424.90	424.90	3.91	0.22	0.00	0.60	3.99	0.21	0.00	0.59	**1.61**	0.25	0.00	0.33
Gas67-22x5	585.11	585.11	0.28	0.14	0.00	0.20	0.10	0.04	0.00	0.07	**0.10**	0.05	0.00	0.23
Min92-27x5	3062.02	3062.02	5.62	0.27	0.00	0.80	5.62	0.29	0.00	0.73	**0.00**	0.21	0.00	0.31
Gas67-29x5	512.10	512.10	4.72	0.41	0.00	1.00	4.89	0.46	0.00	1.01	**1.88**	0.44	0.00	4.20
Gas67-32x5	562.22	562.22	6.11	0.61	0.00	3.45	5.72	0.51	0.00	1.76	**1.24**	0.57	0.00	6.52
Gas67-32x5-2	504.33	504.33	3.46	0.39	0.00	0.50	3.27	0.80	0.00	1.01	**0.01**	0.51	0.00	0.70
Gas67-36x5	460.37	460.37	2.79	0.72	0.00	2.10	1.30	1.40	0.00	2.80	**0.00**	1.04	0.00	1.04
Chr69-50x5ba	565.62	565.62					5.62	3.74	0.00	44.78	**1.58**	6.53	0.00	12.08
Chr69-50x5be	565.60	565.60	10.15	2.70	0.00	181.10	8.85	3.04	0.00	68.79	**2.14**	8.70	0.00	27.36
Perl83-55x15	1112.06	1112.06					3.42	6.17	0.70	7496.92	**1.96**	197.21	**0.00**	240.75
Chr69-75x10ba	886.30	**844.40**					10.23	23.54	4.51	7408.62	**7.27**	1243.61	**0.00**	7006.42
Chr69-75x10be	848.85	848.85	10.83	37.66	4.48	3017.83	10.42	15.25	3.26	7367.52	**6.71**	485.76	**0.00**	4430.99
Chr69-75x10bmw	802.08	802.08					9.27	20.18	3.37	7468.83	**6.02**	207.56	**0.00**	1940.93
Perl83-85x7	1622.50	1622.50					2.53	18.93	0.68	7557.62	**1.65**	76.12	**0.00**	177.92
Chr69-100x10	833.43	833.43					4.89	9.71	0.51	7385.58	**1.81**	419.35	**0.00**	1549.77
Average BBPPW[1]			5.32	4.79	0.50	356.40	4.91	2.44	0.36	827.14	**1.52**	55.28	**0.00**	496.85
Average CCG-BC[2]							5.05	6.52	0.81	2800.42	**2.12**	165.50	**0.00**	962.47

[1] Average on instances reported by Belenguer et al. (2010).
[2] Average on instances reported by Contardo et al. (2010).

Table 6: Comparison with the methods of Belenguer et al. (2010) and Contardo et al. (2010) on family $\mathcal{F}_1$

Instance	z_{UB}	z^*	BBPPW				CCG-BC				CCG-BCP			
			gap_{lr}	t_{lr}	gap	t	gap_{lr}	t_{lr}	gap	t	gap_1	t_1	gap	t
ppw-20x5-0a	54793	54793	7.13	0.34	0.00	2.41	4.57	0.35	0.00	5.04	**3.06**	0.29	0.00	0.60
ppw-20x5-0b	39104	39104	0.00	0.17	0.00	0.13	0.00	0.02	0.00	0.03	0.00	0.03	0.00	0.03
ppw-20x5-2a	48908	48908	3.53	0.25	0.00	2.81	2.71	0.26	0.00	1.31	**2.40**	0.14	0.00	0.56
ppw-20x5-2b	37542	37542	0.00	0.06	0.00	0.06	0.00	0.01	0.00	0.02	0.00	0.02	0.00	0.02
ppw-50x5-0a	90111	90111	11.61	12.38	2.26	7212.25	10.94	27.10	1.77	7336.51	**5.95**	7.25	**0.00**	17.08
ppw-50x5-0b	63242	63242	7.96	3.97	1.37	5557.95	7.50	5.05	1.23	7304.54	**3.55**	4.18	**0.00**	478.03
ppw-50x5-2a	88298	88298	7.48	8.14	1.11	7208.06	7.52	5.08	1.00	7265.57	**3.92**	6.31	**0.00**	13.49
ppw-50x5-2b	67340	**67308**	5.19	2.45	1.43	6013.58	5.63	2.75	1.17	7282.73	**3.77**	2.84	**0.00**	444.33
ppw-50x5-2a	84055	84055	1.97	7.19	0.47	7207.95	**1.95**	29.50	0.28	7244.32	1.99	7.44	**0.00**	29.53
ppw-50x5-2b	51822	51822	**0.67**	1.55	0.00	9.16	0.86	1.76	0.00	10.64	0.69	1.49	0.00	7.25
ppw-50x5-3a	86203	86203	11.25	6.88	1.80	7206.95	10.23	14.67	1.16	7283.89	**3.93**	13.19	**0.00**	94.24
ppw-50x5-3b	61830	61830	7.95	3.14	0.00	96.86	6.26	4.38	0.00	71.76	**2.30**	5.78	0.00	86.73
ppw-100x5-0a	274814	274814					**3.56**	2509.03	2.36	7293.80	4.55	144.98	**0.00**	446.13
ppw-100x5-0b	214392	**213568**					3.21	391.48	2.19	7420.00	**3.14**	119.05	**0.29**	47846.20
ppw-100x5-2a	193671	193671					3.77	365.93	1.60	7398.86	**3.52**	36.06	**0.00**	56.53
ppw-100x5-2b	157173	**157095**					2.34	83.27	0.78	7323.10	**2.14**	42.68	**0.00**	12576.30
ppw-100x5-3a	200079	200079					8.82	108.07	1.44	7340.95	**3.55**	35.56	**0.00**	123.74
ppw-100x5-3b	152441	152441					5.08	27.40	0.57	7332.99	**1.95**	23.85	**0.00**	469.89
ppw-100x10-0a	289017	289017					7.88	1133.84	3.74	7394.33	**5.77**	1147.03	**1.27**	46343.60
ppw-100x10-0b	234641	234641					4.74	147.20	2.48	7334.20	**4.44**	163.75	**1.94**	38092.70
ppw-100x10-2a	243590	243590					4.07	1473.84	1.40	7351.82	**2.82**	169.38	**0.00**	2252.91
ppw-100x10-2b	203988	203988					2.50	90.42	0.00	4734.83	**0.79**	72.55	0.00	324.72
ppw-100x10-3a	252421	252421					8.65	740.38	4.02	7326.18	**6.02**	1425.99	**1.60**	24913.40
ppw-100x10-3b	204597	204597					5.00	112.22	2.15	7338.32	**4.02**	150.36	**1.36**	80196.10
ppw-200x10-1a	479425	478845									**8.66**	3861.00	**1.15**	49799.00
ppw-200x10-1b	378773	378773									**5.32**	3367.07	**1.19**	237100.00
ppw-200x10-2a	450468	450468									**5.06**	359.83	**0.80**	31241.20
ppw-200x10-2b	374435	374435									**3.10**	566.81	**0.37**	92495.60
ppw-200x10-3a	472898	**469433**									**6.61**	3788.52	**0.00**	18738.80
ppw-200x10-3b	364178	364178									**4.92**	2482.11	**1.00**	45013.90
Average BBPPW[1]			5.39	3.88	0.70	3376.51	4.85	7.58	0.55	3650.53	**2.63**	4.08	**0.00**	97.66
Average CCG-BC[2]							4.91	303.08	1.22	5391.49	**3.09**	149.17	**0.27**	10617.25

[1] Average on instances reported by Belenguer et al. (2010).
[2] Average on instances reported by Contardo et al. (2010).

Table 7: Comparison with the methods of Belenguer et al. (2010) and Contardo et al. (2010) on family $\mathcal{F}_2$

Instance	z_{UB}	z^*	BBPPW				CCG-BC				CCG-BCP			
			gap_{lr}	t_{lr}	gap	t	gap_{lr}	t_{lr}	gap	t	gap_1	t_1	gap	t
r30x5a-1	819.51	819.51	4.28	0.70	0.00	50.22	3.33	0.89	0.00	3.23	**2.89**	0.60	0.00	2.45
r30x5a-2	821.50	821.46	6.41	0.53	0.00	53.89	5.89	0.41	0.00	8.77	**3.73**	0.38	0.00	3.72
r30x5a-3	702.30	702.29	1.09	0.52	0.00	0.73	0.56	0.71	0.00	0.91	**0.00**	0.44	0.00	0.50
r30x5b-1	880.02	880.02	7.58	0.47	0.00	8.48	7.39	0.52	0.00	9.05	**2.69**	1.01	0.00	4.57
r30x5b-2	825.30	825.30	4.38	0.50	0.00	1.09	3.52	1.31	0.00	2.55	**1.22**	0.97	0.00	1.24
r30x5b-3	884.60	884.58	3.14	0.95	0.00	5.63	3.33	1.09	0.00	3.25	**2.33**	0.92	0.00	1.23
r40x5a-1	928.10	928.10	9.32	1.14	0.00	305.25	8.95	1.32	0.00	140.31	**3.30**	3.99	0.00	14.67
r40x5a-2	888.40	888.40	8.86	0.94	0.00	98.34	8.83	1.04	0.00	86.31	**2.80**	2.25	0.00	11.88
r40x5a-3	947.30	947.30	7.66	2.34	0.00	158.27	7.47	2.48	0.00	76.63	**3.02**	3.09	0.00	11.36
r40x5b-1	1052.00	1052.00	10.60	1.31	0.00	3694.45	10.26	2.80	0.00	3115.92	**5.78**	5.49	0.00	10.49
r40x5b-2	981.50	981.50	8.92	1.38	0.00	10.25	8.57	1.26	0.00	7.61	**2.09**	1.63	0.00	3.77
r40x5b-3	964.30	964.30	5.21	1.48	0.00	11.36	4.51	2.32	0.00	12.33	**2.00**	1.67	0.00	2.68
Average			6.45	1.02	0.00	366.50	6.05	1.35	0.00	288.91	**2.65**	1.87	0.00	5.71

Table 8: Comparison with the methods of Belenguer et al. (2010) and Contardo et al. (2010) on family $\mathcal{F}_3$

Instance	z_{UB}	z^*	BMW							CCG-BCP						
			gap_1	t_1	gap_2	t_2	gap_3	t_3	t	gap_1	t_1	gap_2	t_2	gap_3	t_3	t
Perl-12x2	203.98	203.98	1.50	0.30	0.00	0.20	0.00	0.00	0.50	**0.00**	0.02	0.00	0.01	0.00	0.00	0.03
Gas-21x5	424.90	424.90	2.40	3.10	0.00	0.80	0.00	0.00	3.90	**1.61**	0.25	0.00	0.08	0.00	0.00	0.33
Gas-22x5	585.11	585.11	1.50	5.40	0.00	0.60	0.00	0.00	6.00	**0.10**	0.05	0.00	0.18	0.00	0.00	0.23
Min-27x5	3062.02	3062.02	3.00	39.10	0.00	7.90	0.00	0.00	47.00	**0.00**	0.21	0.00	0.10	0.00	0.00	0.31
Gas-29x5	512.10	512.10	7.20	110.70	0.00	67.50	0.00	0.00	178.20	**1.88**	0.44	0.00	3.76	0.00	0.00	4.20
Gas-32x5	562.22	562.22	6.00	13.00	0.10	45.60	0.00	4.80	63.40	**1.24**	0.57	**0.00**	5.95	0.00	0.00	6.52
Gas-32x5b	504.33	504.33	2.60	99.60	0.00	18.30	0.00	0.00	117.90	**0.01**	0.51	0.00	0.19	0.00	0.00	0.70
Gas-36x5	460.37	460.37	5.50	1.60	0.00	1.30	0.00	0.00	2.90	**0.00**	1.04	0.00	0.00	0.00	0.00	1.04
Chr-50x5ba	565.62	565.62	5.80	48.90	0.00	44.50	0.00	0.50	93.90	**1.58**	6.53	0.00	5.55	0.00	0.00	12.08
Chr-50x5be	565.60	565.60	6.00	47.10	0.00	65.80	0.00	0.00	112.90	**2.14**	8.70	0.00	18.66	0.00	0.00	27.36
Perl-55x15	1112.06	1112.06	3.10	102.20	0.00	189.00	0.00	0.00	291.20	**1.96**	197.21	0.00	43.54	0.00	0.00	240.75
Chr-75x10ba	886.30	**844.40**								**7.27**	1243.61	**0.48**	5749.77	**0.00**	13.04	7006.42
Chr-75x10be	848.85	848.85	7.80	1330.40	0.10	2072.60	0.00	10.50	3413.50	**6.71**	485.76	**0.00**	3945.23	0.00	0.00	4430.99
Chr-75x10bmw	802.08	802.08	7.80	1004.70	0.50	790.30	0.00	1031.90	2826.90	**6.02**	207.56	**0.00**	1733.37	0.00	0.00	1940.93
Perl-85x7	1622.50	1622.50	2.80	221.90	0.00	266.20	0.00	0.00	488.10	**1.65**	76.12	0.00	101.80	0.00	0.00	177.92
Chr-100x10	833.43	833.43	6.80	2609.90	0.30	9898.60	0.00	566.20	13074.70	**1.81**	419.35	**0.00**	1130.42	0.00	0.00	1549.77
Average BMW[1]			4.65	375.86	0.07	897.95	0.00	107.59	1381.40	**2.12**	165.50	**0.03**	796.16	0.00	0.81	962.47

[1] Average on instances reported by Baldacci et al. (2010b).

Table 9: Comparison with the method of Baldacci et al. (2010b) on family $\mathcal{F}_1$

Instance	z_{UB}	z^*	BMW							CCG-BCP						
			gap_1	t_1	gap_2	t_2	gap_3	t_3	t	gap_1	t_1	gap_2	t_2	gap_3	t_3	t
ppw-20x5-0a	54793	54793	3.60	8.00	0.10	2.10	0.00	0.10	10.20	**3.06**	0.32	**0.00**	0.32	0.00	0.00	0.64
ppw-20x5-0b	39104	39104	2.10	8.70	0.00	9.20	0.00	0.00	17.90	**0.00**	0.03	0.00	0.00	0.00	0.00	0.03
ppw-20x5-2a	48908	48908	**0.80**	1.60	0.00	2.20	0.00	0.00	3.80	2.40	0.16	0.00	0.33	0.00	0.00	0.49
ppw-20x5-2b	37542	37542	3.70	12.10	0.00	32.70	0.00	0.00	44.80	**0.00**	0.02	0.00	0.00	0.00	0.00	0.02
ppw-50x5-0a	90111	90111	**5.90**	45.80	0.30	6.70	0.00	0.40	52.90	5.96	20.50	**0.11**	12.88	0.00	0.10	33.48
ppw-50x5-0b	63242	63242	5.10	467.10	2.10	92.90	0.00	8368.90	8928.90	**4.01**	9.00	**0.58**	648.79	0.00	503.83	1161.62
ppw-50x5-2a	88298	88298	6.10	8.40	1.40	10.50	0.00	52.70	71.60	**4.05**	11.39	**0.07**	6.31	0.00	0.05	17.75
ppw-50x5-2b	67340	**67308**	6.10	69.00	2.70	75.90	2.70	9386.90	9531.80	**3.77**	3.70	**0.00**	227.31	**0.00**	0.00	231.01
ppw-50x5-2a	84055	84055	4.00	7.80	0.60	20.50	0.00	30.20	58.50	**2.01**	4.48	**0.01**	27.30	0.00	0.08	31.86
ppw-50x5-2b	51822	51822	6.50	55.90	0.00	136.10	0.00	0.00	192.00	**0.69**	1.99	0.00	9.79	0.00	0.00	11.78
ppw-50x5-3a	86203	86203	6.10	20.20	1.00	18.80	0.00	22.50	61.50	**3.92**	33.37	**0.82**	30.94	0.00	107.65	171.96
ppw-50x5-3b	61830	61830	5.50	45.00	0.30	80.40	0.00	5.80	131.20	**2.53**	8.29	**0.00**	74.80	0.00	0.00	83.09
ppw-100x5-0a	274814	274814	**1.20**	292.30	0.20	63.70	0.00	46.60	402.60	4.55	514.97	0.20	322.41	0.00	133.69	971.07
ppw-100x5-0b	214392	**213568**	**0.72**	773.60	**0.42**	91.00	0.42	8869.60	9734.20	3.14	391.58	0.43	6858.13	0.29	12354.60	19604.40
ppw-100x5-2a	193671	193671	**1.30**	91.10	0.10	23.10	0.00	2.30	116.50	3.52	112.82	**0.06**	22.22	0.00	0.66	135.70
ppw-100x5-2b	157173	**157095**	**1.80**	2419.30	0.40	624.20	0.40	12415.40	15458.90	2.14	146.75	**0.10**	15324.80	**0.00**	5392.35	20863.90
ppw-100x5-3a	200079	200079	**2.10**	227.00	0.20	31.70	0.00	14.70	273.40	3.59	118.87	**0.17**	67.38	0.00	38.95	225.20
ppw-100x5-3b	152441	152441	**2.00**	734.20	0.10	270.50	0.00	14.80	1019.50	2.08	70.44	**0.00**	539.20	0.00	0.00	609.64
ppw-100x10-0a	289017	289017	**2.70**	257.50	1.90	115.60	1.90	23089.40	23462.50	5.76	4411.79	**1.69**	1627.38	**1.35**	42053.90	48093.10
ppw-100x10-0b	234641	234641	**3.20**	426.60	2.20	437.60	2.20	19278.00	20142.20	4.39	581.21	**2.08**	20735.90	**2.04**	22541.80	43858.80
ppw-100x10-2a	243590	243590	**2.60**	275.70	0.50	65.50	0.00	7495.60	7836.80	2.82	589.95	**0.39**	725.90	0.00	5677.72	6993.57
ppw-100x10-2b	203988	203988	1.90	842.20	0.10	882.20	0.00	31.50	1755.90	**0.80**	210.36	**0.00**	339.55	0.00	0.00	549.91
ppw-100x10-3a	252421	252421	6.20	100.50	2.10	137.00	2.10	14558.70	14796.20	**6.02**	4011.62	**1.91**	1231.36	**1.53**	33510.20	38753.20
ppw-100x10-3b	204597	204597	4.40	504.10	1.60	529.70	1.60	19289.50	20323.30	**4.00**	616.12	**1.55**	15007.20	**1.53**	23181.40	38804.70
ppw-200x10-1a	479425	478845								**8.55**	12970.70	**1.15**	12418.10	**1.04**	32031.80	57420.70
ppw-200x10-1b	378773	378773								**5.32**	11168.90	**1.18**	383307.00	**1.18**	157438.00	551914.00
ppw-200x10-2a	450468	450468								**5.06**	1107.24	**0.87**	9552.10	**0.82**	21913.90	32573.20
ppw-200x10-2b	374435	374435								**3.04**	1509.01	**0.40**	68446.80	**0.36**	76737.80	146694.00
ppw-200x10-3a	472898	**469433**								**6.61**	13541.20	**0.13**	11677.20	**0.00**	11579.70	36798.00
ppw-200x10-3b	364178	364178								**4.92**	9072.59	**1.01**	56865.70	**1.01**	15068.20	81006.50
Average BMW[1]			3.57	320.57	0.76	156.66	0.47	5123.90	5601.13	**3.13**	494.57	**0.42**	2660.01	**0.28**	6062.37	9216.95
Average on solved by BMW			3.56	184.89	0.41	102.87	0.00	946.24	1234.00	**2.71**	100.41	**0.14**	166.36	**0.00**	380.16	646.93
Average on solved by CCG-BCP			3.60	296.39	0.53	128.89	0.16	1994.13	2419.41	**2.73**	97.76	**0.13**	967.38	**0.00**	623.95	1689.09

[1] Average on instances reported by Baldacci et al. (2010b).

Table 10: Comparison with the method of Baldacci et al. (2010b) on family $\mathcal{F}_2$

Instance	z_{UB}	z^*	BMW							CCG-BCP						
			gap_1	t_1	gap_2	t_2	gap_3	t_3	t	gap_1	t_1	gap_2	t_2	gap_3	t_3	t
r30x5a-1	819.5	819.5	3.30	49.20	0.70	24.80	0.00	1.40	75.40	**2.89**	0.60	**0.08**	1.85	0.00	0.02	2.47
r30x5a-2	821.5	821.5	5.40	88.60	1.60	27.50	0.00	5.80	121.90	**3.73**	0.38	**0.00**	3.34	0.00	0.00	3.72
r30x5a-3	702.3	702.3	3.70	35.80	0.00	30.50	0.00	0.00	66.30	**0.00**	0.44	0.00	0.06	0.00	0.00	0.50
r30x5b-1	880.0	880.0	6.40	75.40	0.00	21.50	0.00	0.00	96.90	**2.69**	1.01	0.00	3.56	0.00	0.00	4.57
r30x5b-2	825.3	825.3	3.00	50.30	0.00	6.80	0.00	0.00	57.10	**1.22**	0.97	0.00	0.27	0.00	0.00	1.24
r30x5b-3	884.6	884.6	**1.30**	31.90	0.00	7.30	0.00	0.00	39.20	2.33	0.92	0.00	0.31	0.00	0.00	1.23
r40x5a-1	928.1	928.1	6.60	169.60	0.00	99.50	0.00	0.00	269.10	**3.30**	3.99	0.00	10.68	0.00	0.00	14.67
r40x5a-2	888.4	888.4	5.60	181.20	0.20	78.80	0.00	0.70	260.70	**2.80**	2.25	**0.19**	9.58	0.00	0.05	11.88
r40x5a-3	947.3	947.3	4.90	158.80	0.10	84.50	0.00	0.80	244.10	**3.02**	3.09	**0.00**	8.27	0.00	0.00	11.36
r40x5b-1	1052.0	1052.0	**5.50**	159.70	0.00	70.60	0.00	0.00	230.30	5.78	5.49	0.00	5.00	0.00	0.00	10.49
r40x5b-2	981.5	981.5	6.20	213.90	0.00	82.90	0.00	0.00	296.80	**2.09**	1.63	0.00	2.14	0.00	0.00	3.77
r40x5b-3	964.3	964.3	3.30	209.70	0.00	21.50	0.00	0.00	231.20	**2.00**	1.67	0.00	1.01	0.00	0.00	2.68
Average			4.60	118.67	0.22	46.35	0.00	0.72	165.75	**2.65**	1.87	**0.02**	3.84	0.00	0.01	5.71

Table 11: Comparison with the method of Baldacci et al. (2010b) on family $\mathcal{F}_3$

Instance	z_{UB}	z^*	BMW							CCG-BCP						
			gap_1	t_1	gap_2	t_2	gap_3	t_3	t	gap_1	t_1	gap_2	t_2	gap_3	t_3	t
P111112	1467.68	1467.68	8.70	1471.60	0.20	3039.80	0.00	57.60	4569.00	**4.63**	885.82	**0.00**	3581.25	0.00	0.00	4467.07
P111212	1394.80	1394.80	9.00	1571.30	0.40	5122.90	0.00	416.30	7110.50	**4.67**	859.13	**0.00**	20689.10	0.00	0.00	21548.23
P112112	1167.16	1167.16	4.70	1518.40	0.00	2503.50	0.00	0.00	4021.90	**2.74**	75.47	0.00	235.30	0.00	0.00	310.77
P112212	791.66	791.66	6.00	1818.90	0.10	4100.60	0.00	15.50	5935.00	**1.36**	21.32	**0.00**	1818.15	0.00	0.00	1839.47
P113112	1245.45	**1238.24**	9.10	2615.30	1.60	17579.50	1.60	37716.50	57911.30	**3.95**	454.58	**0.61**	118970.00	**0.34**	52278.40	171702.98
P113212	902.26	902.26	5.20	2755.00	0.00	4509.60	0.00	0.00	7264.60	**0.49**	51.30	0.00	101.93	0.00	0.00	153.23
P131112	1900.70	**1892.17**	7.50	2408.70	1.00	7156.60	1.00	27217.40	36782.70	**6.24**	11926.90	**0.85**	299823.00	**0.64**	38047.20	349797.10
P131212	1965.12	1965.12	**8.00**	2165.30	1.00	6686.30	1.00	17537.90	26389.50	8.43	10682.50	**0.95**	207942.00	**0.82**	33676.30	252300.80
P132112	1443.33	1443.32	**3.00**	19150.40	0.00	19081.70	0.00	70.90	38303.00	5.68	3582.12	0.00	15021.90	0.00	0.00	18604.02
Average			6.80	3941.66	0.48	7753.39	0.40	9225.79	20920.83	**4.24**	3171.02	**0.27**	74242.51	**0.20**	13777.99	91191.52
Average on solved instances			6.10	4714.27	0.12	6393.02	0.00	93.38	11200.67	**3.26**	912.53	**0.00**	6907.94	0.00	0.00	7820.47

Table 12: Comparison with the method of Baldacci et al. (2010b) on family $\mathcal{F}_4$

Instance	z_{UB}	z^*	BMW							CCG-BCP						
			gap_1	t_1	gap_2	t_2	gap_3	t_3	t	gap_1	t_1	gap_2	t_2	gap_3	t_3	t
M-n150x14a	1352.93	1352.93	**7.60**	1266.20	0.20	89144.50	0.00	5320.10	95730.80	9.61	34342.60	**0.09**	351540.00	0.00	1.32	385883.92
M-n150x14b	1212.46	1212.46	**7.30**	2499.80	0.40	48694.00	0.00	324.00	51517.80	7.50	39786.80	**0.20**	352454.00	0.00	30.68	392271.48
M-n199x14a	1644.35	1644.35	**6.50**	14428.10	0.30	188049.90	0.00	606.50	203084.50	12.06	64412.40	**0.15**	616522.00	0.00	10.81	680945.21
M-n199x14b	1480.43	1480.43	**7.40**	6187.60	0.10	259498.30	0.00	149.60	265835.50	10.22	67992.50	**0.09**	1006620.00	0.00	1.98	1074614.48
Average			**7.20**	6095.43	0.25	146346.67	0.00	1600.05	154042.15	9.85	51633.57	**0.13**	581784.00	0.00	11.20	633428.77

Table 13: Comparison with the method of Baldacci et al. (2010b) on family $\mathcal{F}_5$

References

Z. Akca, R. T. Berger, and T. K. Ralphs. Modeling and solving location, routing, and scheduling problems. In *Proceedings of the Eleventh INFORMS Computing Society Meeting*, pages 309–330, Charleston, South Carolina, USA, 2009.

C. Archetti, M. Bouchard, and G. Desaulniers. Enhanced branch-and-price-and-cut for vehicle routing with split deliveries and time windows. Technical Report G-2009-81, Cahiers du GERAD, Montréal, Canada, 2009.

P. Augerat. *Approche polyhédrale du problème de tournées de véhicules*. PhD thesis, Institut National Polytechnique de Grenoble, France, 1995.

E. Balas and M. W. Padberg. Set-partitioning: a survey. *SIAM Review*, 18(4):710–760, 1976.

R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351 – 385, 2008.

R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti. An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, 7:229–268, 2010a.

R. Baldacci, A. Mingozzi, and R. Wolfler-Calvo. An exact method for the capacitated location-routing problem. *Operations Research*, 2010b. Forthcoming.

S. Barreto. *Análise e Modelização de Problemas de localização-distribuição*. PhD thesis, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal. In Portuguese., 2004.

J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler-Calvo. A branch-and-cut algorithm for the location routing problem with depot and vehicle capacities. *Computers & Operations Research*, 38:931–941, 2010.

A. Billionet, S. Elloumi, and L. Grouz Djerbi. Designing radio-mobile access networks based on synchronous digital hierarchy rings. *Computers & Operations Research*, 32(2):379–394, 2005.

B. Boland, J. Dethridge, and I. Dumitrescu. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34:58–68, 2006.

N. Christofides, A. Mingozzi, and P. Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20:255–282, 1981.

C. Contardo, J.-F. Cordeau, and B. Gendron. A computational comparison of flow formulations for the capacitated location-routing problem. Technical report, CIRRELT, 2010.

G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999.

D. Feillet, M. Gendreau, and L.-M. Rousseau. New refinements for the solution of vehicle routing problems with branch and price. *Infor*, 45(4):239–256, 2007.

R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, and R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming Series A*, 106:491–511, 2006.

H. Gunnarsson, M. Rönnqvist, and D. Carlsson. A combined terminal location and ship routing problem. *Journal of the Operational Research Society*, 57:928–938, 2006.

P. H. Hansen, B. Hegedahl, S. Hjortkjaer, and B. Obel. A heuristic solution to the warehouse location-routing problem. *European Journal of Operational Research*, 76:111–127, 1994.

M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511, 2008.

M. Labbé and G. Laporte. Maximizing user convenience and postal service efficiencyin post box location. *Journal of Operational Research, Statistics and Computer Science*, 26, 1986.

C. K. Y. Lin, C. K. Chow, and A. Chen. A location-routing-loading problem for bill delivery services. *Computers & Operations Research*, 43:5–25, 2002.

S. C. Liu and C. C. Lin. A heuristic method for the combined location routing and inventory problem. *International Journal of Advanced Manufacturing Technologies*, 26:372–381, 2005.

J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100:423–445, 2004.

J. Perl and M. S. Daskin. A warehouse location-routing problem. *Transportation Research B*, 19B: 381–396, 1985.

C. Prins, C. Prodhon, and R. Wolfer Calvo. A memetic algorithm with population management (ma-pm) for the capacitated location-routing problem. *Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science*, 3906:183–194, 2006a.

C. Prins, C. Prodhon, and R. Wolfer Calvo. Solving the capacitated location-routing problem by a grasp complemented by a learning process and path relinking. *4OR*, 4(3):221–238, 2006b.

C. Prins, C. Prodhon, A. Ruiz, P. Soriano, and R. Wolfer Calvo. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, 41(4):470–483, 2007.

G. Righini and M. Salani. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51:155–170, 2008.

D. Tuzun and L. I. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116:87–99, 1999.

T.-H. Wu, C. Low, and J.-W. Bai. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, 29:1393–1415, 2002.