Non-Robust Strong Knapsack Cuts for Capacitated Location-Routing and Related Problems

Pedro Liguori, Ali Ridha Mahjoub, Guillaume Marques, Ruslan Sadykov, Eduardo Uchoa

Non-Robust Strong Knapsack Cuts for Capacitated Location-Routing and Related Problems

Pedro Henrique Liguori*[1], A. Ridha Mahjoub†[1], Guillaume Marques‡[2], Ruslan Sadykov§[3], and Eduardo Uchoa¶[4]

[1]LAMSADE CNRS UMR 7243, Université Paris-Dauphine PSL Research University, Paris, 75016, France
[2]Atoptima, Bordeaux, France
[3]EDGE, Inria Bordeaux Sud–Ouest, Bordeaux, France
[4]Engenharia de Produção, University Federal Fluminense, Niteroi-RJ, 24210-240, Brazil

2 December 2022

Abstract

The Capacitated Location-Routing Problem consists in, given a set of locations and a set of customers, determining in which locations one should install depots with limited capacity, and for each depot, design a number of routes to supply customer demands. We provide a formulation that includes depot variables, edge variables, assignment variables, and an exponential number of route variables, together with some new families of valid inequalities, leading to a branch-cut-and-price algorithm. The main original methodological contribution of the article is the Route Load Knapsack Cuts, a family of non-robust cuts, defined over the route variables, devised to strengthen the depot capacity constraints. We explore the monotonicity and the superadditivity properties of those cuts to adapt the labeling algorithm, used in the pricing, for handling the additional dual variables efficiently. Computational experiments show that several Capacitated Location-Routing previously unsolved instances from the literature can now be solved to optimality. Additional experiments with hard instances of the Vehicle Routing Problem with Capacitated Multiple Depots and with instances of the Vehicle Routing Problem with Time Windows and Shifts indicate that the newly proposed cuts are also effective for those problems.

1 Introduction

Location-Routing Problems (LRPs) arise when combining two classic combinatorial optimization problems: facility location and vehicle routing. In fact, the integration of both levels of decisions, *i.e.*, depot location and vehicle routing, makes the LRP an interesting model for several practical applications, from the design of telecommunications networks to the operation of very competitive supply chains. Making decisions on the location of depots and the routing of vehicles independently usually leads to strongly suboptimal planning results, as observed by Salhi and Rand (1989). As a result, LRPs have been extensively studied in the literature, as surveyed in

*phliguori@gmail.com
†ridha.mahjoub@lamsade.dauphine.fr
‡guillaume.marques@atoptima.com
§ruslan.sadykov@inria.fr
¶eduardo_uchoa@id.uff.br

Schneider and Drexl (2017). The importance of LRPs is currently rising due to the surge of home delivery services and e-commerce. In those contexts, solving LRPs help in determining the location of urban depots from which customers would be served on vehicle routes.

The Capacitated LRP (CLRP) is defined as follows. Consider an undirected graph $G = (V, E)$, with $V = I \cup J$ and $E = E_{IJ} \cup E_J$, where nodes in I represent possible depot locations, nodes in J denote customers, E_J is the set of all pairs of distinct nodes in J, and $E_{IJ} = I \times J$. Note that subgraph (J, E_J) is a complete graph, subgraph $(I \cup J, E_{IJ})$ is a complete bipartite graph, and vertices of I form an independent set. For each edge $e \in E$, there is a cost $c_e \in \mathbb{R}_+$. For each depot location $i \in I$, there is an opening cost $f_i \in \mathbb{R}_+$ and a capacity $W_i \in \mathbb{Z}_+$. For each customer $j \in J$, there is a demand $d_j \in \mathbb{Z}_+$. Finally, there is an unlimited number of identical vehicles with capacity $Q \in \mathbb{Z}_+$. A *route* is a cycle in G passing through exactly one depot of I and through a set of distinct customers in J having total demand not exceeding Q. We say that a route *leaves* (or *is incident to*) the depot i in I that it contains. A CLRP solution is a set of opened depots $I' \subset I$ and a set of routes such that: (i) all routes leave opened depots, (ii) all customers are visited exactly once, and (iii) the sum of the demands in all routes leaving a depot $i \in I'$ should not exceed W_i. The goal is to minimize the sum of the opening costs plus the cost of the edges in the routes. We remark that we are assuming that there are no split deliveries, so each customer should be indeed visited only once. That modeling assumption is reasonable because the edges in G actually represent shortest paths between pairs of points, obtained from a street/road network. If the shortest path between A and B happens to pass by the street segment in front of customer C, this is not counted as a visit to C.

A closely related problem, also considered in this article, is the Vehicle Routing Problem with Capacitated Multiple Depots (VRP-CMD). It differs from CLRP only by having zero opening costs, so all depots can be considered as opened. The third problem addressed here is the Vehicle Routing Problem with Time Windows and Shifts (VRPTW-S), introduced in Dabia et al. (2019). The problem is a generalization of the classic VRPTW, in which we have a single depot, which is already open. In addition to demands, every customer $j \in J$ also has a time window $[l_j, u_j]$ during which it should be visited. In the VRPTW-S, there is a set I of shifts. Each shift $i \in I$ is characterized by a time window $[l_i, u_i]$ during which a vehicle may leave the depot, and a capacity W_i limiting the total load of vehicles departing during the shift. To reduce the VRPTW-S to the LRP with time windows, we introduce a fictive depot for every shift. The position of all fictive depots is the same as the original depot. However, a time window for leaving every fictive depot $i \in I$ is set to $[l_i, u_i]$. As for the VRP-CMD, all fictive depots are considered to be open.

All those problems have a *nested knapsack* structure. The customers are first assigned to routes that have the following knapsack-like constraints: the sum of the demands (the load) should not exceed vehicle capacity. Then the routes themselves are attached to depots/shifts subject to knapsack-like constraints: the sum of the loads should not exceed depot/shift capacity. Similar nested knapsack structures are encountered in several other problems, some of which are mentioned in the literature review in Section 2.

Branch-Cut-and-Price (BCP) algorithms are the best existing methods for the exact solution of most vehicle routing variants (see, e.g., Poggi and Uchoa (2014) and Costa et al. (2019)). A crucial issue in that kind of algorithm is the impact of cut separation on the pricing. According to the classification proposed in Poggi and Uchoa (2003), a family of cuts is *robust* if they do not change the structure of the pricing subproblem. A BCP algorithm is said to be to robust if it only uses robust cuts. A prototypical example of a robust BCP algorithm for VRP is Fukasawa et al. (2006). The robust VRP cuts are expressed over the edge/arc variables of a suitable original compact formulation, they are translated to the route variables using a linear mapping. In that way, the dual variables of the separated cuts only change edge/arc costs in the pricing. On the other hand, cuts that can only be expressed directly over the route variables are *non-robust*. Each separated cut adds an extra resource to the labeling algorithm used in the pricing, making it harder.

Non-robust cuts are often stronger than robust ones precisely because they have access to full route information instead of only having access to "small bits of routes" (edge/arc variables). In

fact, non-robust cuts for VRPs often explore the set-partitioning constraints that state that each customer should be visited by exactly one route. Classic set-partitioning cuts (like the Clique cuts used in Baldacci et al. (2008)), in spite of being potentially very strong, are not really practical: a few dozen such cuts already makes the pricing intractable. The Subset Row cuts, introduced in Jepsen et al. (2008), are a family of set-partitioning cuts purposely created to be used in a BCP context, designed to be less harmful to the pricing subproblem. An important advance was the introduction of *limited-memory cuts* (Pecin et al., 2014), a technique (first used on Subset Row cuts and later generalized to Rank-1 cuts in Pecin et al. (2017b)) to dynamically adjust cut coefficients in order to substantially reduce their negative impact on the pricing. In spite of those improvements, those limited-memory cuts are still non-robust and should be handled with care. Indeed, the BCPs in Pecin et al. (2014) and Pessoa et al. (2020) use a roll-back mechanism to remove active non-robust cuts (of course, losing bound quality) when the pricing becomes too expensive.

This work proposes a BCP algorithm for the CLRP, that is easily adapted to the VRP-CMD and the VRPTW-S. The algorithm is based on a formulation of the problem that includes, besides depot variables, assignment variables and edge variables, an exponential number of route variables. The BCP also uses robust cuts (including two newly proposed families) and non-robust limited-memory Rank-1 cuts. However, the main methodological advance is the *Route Load Knapsack Cuts* (RLKCs), a new family of non-robust cuts derived from the constraints that state that the sum of loads of the routes attached to a depot should not exceed its capacity.

Dabia et al. (2019) recently proposed four families of non-robust cover cuts (y-cover, k-cover, p-cover, and q-cover) for the similar nested knapsack structure found in VRPTW-S. The newly proposed RLKCs have the following advantages over those cover cuts:

- They are more general and stronger. The cuts in Dabia et al. (2019) are inspired by classic knapsack cover cuts and only have coefficients 0 or 1 on their left-hand side. In contrast, RLKCs include all the facets of the so-called Master Knapsack Polyhedron and can have many distinct coefficients on their left-hand side.
- The cover cuts in Dabia et al. (2019) have a significant negative effect on the performance of the proposed labeling algorithm used in their pricing. In fact, those authors limit the number of active such cuts to 50 in their BCP algorithm, in order to keep the pricing tractable. In contrast, we present a modified labeling algorithm that, by exploring monotonicity and superadditivity properties, can handle RLKCs in a very efficient way. Indeed, our BCP algorithm freely separates RLKCs and does not need any special mechanism to control them.

The remainder of the paper is organized as follows. Section 2 reviews the literature on the CLRP and on some related problems with the nested knapsack structure. Section 3 contains a polynomially sized integer programming formulation for the CLRP, as well as known and new families of valid inequalities for it. Section 4 presents an extension of that formulation, by introducing an exponential number of route variables. Section 5 describes the new RLKCs, also showing how they are separated and how they change the labeling algorithm used for the pricing. Section 6 presents extensive computational experiments on CLRP, VRP-CMD and VRPTW-S. Conclusions and research perspectives are given in Section 7. Detailed instance-by-instance computational results are presented in the appendices.

2 Literature review

The idea of combining two levels of decision, depot location and vehicle routing, is not new. The first exact method for the LRP is due to Laporte and Nobert (1981) in which the authors develop a Branch-and-Cut algorithm for solving a special case where a single depot must be opened among a list of possible depot locations. Afterward, Laporte et al. (1986) investigate the CLRP with multi-depots to be opened and subject to vehicle capacities. The computational

results reported show that they were able to solve instances with 8 depot locations and 20 customers. In a later work, Laporte et al. (1988) discuss the CLRP in a context of asymmetrical costs, where vehicle capacities are replaced by constraints on the maximum length of the routes. Instances are then solved by a Branch-and-Cut algorithm.

Belenguer et al. (2011) use a two-index formulation for the CLRP. By adapting some of the valid inequalities from the CVRP literature, together with others conceived specifically for the CLRP, the authors devise a Branch-and-Cut algorithm. From the computational experiments reported, their approach is able to solve instances with 5 depot locations and 50 customers. Contardo et al. (2013) extend the work of Belenguer et al. (2011) and present four different arc-flow formulations for which they derive several new families of valid inequalities, giving both heuristic and exact separation procedures. The computation results show that a three-index flow formulation is stronger than the two-index counterparts, however, this does not always culminate in a better algorithmic performance.

The first use of column generation on CLRP is due to Berger et al. (2007). Here, the authors develop a Branch-and-Price algorithm to solve instances with uncapacitated depots and routes limited by a maximum length. The authors report computational experiments on instances with 10 depot locations and up to 100 customers, some of which are solved to optimality within a running time of two hours. Akca et al. (2009) give a set partitioning formulation for the standard CLRP and solve it by a branch-and-price algorithm. The authors apply three distinct heuristics to price negative reduced cost columns, calling the exact labeling algorithm only when the heuristics fail to find such columns. They were able to solve instances with up to 5 depot locations and 40 customers.

Baldacci et al. (2011b) propose a solution strategy for the CLRP that consists in solving the VRP-CMD for each possible set of opened depots and keeping the solution with the smallest cost. They propose a sophisticated lower bounding procedure that can reduce a lot the number of tested sets, by identifying many sets of locations that cannot lead to optimal solutions. Their algorithm clearly outperforms the solution methods known at that time.

Contardo et al. (2014) also develop an approach based on the enumeration of subsets of depot locations. A Branch-and-Cut algorithm over the formulation proposed in Belenguer et al. (2011), strengthened by some valid inequalities introduced in Contardo et al. (2013), is used for identifying subsets that can not lead to solutions better than a given upper bound. Then, for each such subset of depots that can possibly lead to improving solutions, the corresponding VRP-CMD instance is solved as follows. First, a strong lower bound is obtained by cut-and-column, then all columns whose reduced costs are not greater than the gap between the upper and lower bounds are enumerated, and finally, the standard set-partitioning formulation having the enumerated columns is solved by a standard MILP solver. The obtained computational results are better than those in Baldacci et al. (2011b), solving two additional instances and providing tighter lower bounds for unsolved instances.

As observed by Schneider and Drexl (2017), both the algorithms proposed by Baldacci et al. (2011b) and Contardo et al. (2014) are very sophisticated and rely on a number of complex algorithmic and implementation refinements. A point worth mentioning is that these methods exploit the fact that the instances then found in the literature have a rather small number of depot locations, at most 10. Hence, the proposed smart enumeration of all subsets of depots is more likely to be manageable. However, it is unclear if those methods could be able to deal with larger instances, such as the new benchmarks introduced by Schneider and Löffler (2019) containing 15, 20 and 30 depot locations.

The CLRP is also a fruitful topic for the development of heuristics. Many of those methods work in a two-stage hierarchical fashion: first decide which depots to open and then optimize the vehicle routing. We refer the reader to the survey of Schneider and Drexl (2017) for a complete overview of heuristic methods for CLRP.

We now review the literature on some related problems with the nested knapsack structure. First of all, the VRP-CMD, the particular case of CLRP where all depots are already opened, still has that structure. As mentioned before, Baldacci et al. (2011b) and Contardo et al.

(2014) have to deal with VRP-CMD subproblems when solving the CLRP. Both works do that by adapting existing approaches, Baldacci and Mingozzi (2009) and Contardo and Martinelli (2014), respectively, for the multi-depot vehicle routing problem with uncapacitated depots. Actually, Contardo et al. (2014) also use valid inequalities proposed by Belenguer et al. (2011) and Contardo et al. (2013). The VRP-CMD is again encountered as a subproblem by Ben Mohamed et al. (2020) when solving the two-echelon stochastic multi-period capacitated location-routing problem by a logic-based Benders decomposition algorithm. Those authors solve the VRP-CMD by a direct adaptation of the BCP algorithm in Sadykov et al. (2021). No specific inequalities for the VRP-CMD are used.

Dabia et al. (2019) introduced the Vehicle Routing Problem with Time Windows and Shifts (VRPTW-S). In this generalization of the class vehicle routing problem with time windows, the time horizon is divided in non-overlapping shifts. Depending of the time when a route leaves from the depot, this route is assigned to one of the shifts. The total amount of freight delivered by routes belonging to a shift is limited by a loading capacity. Thus, the nested knapsack structure appears, as every customer's demand contributes to the vehicle capacity and the shift loading capacity constraints. Dabia et al. (2019) propose a BCP algorithm for the VRPTW-S. Their main contribution concerns new cover inequalities for the problem. These inequalities are related to the RLKCs proposed in this paper, as they are also derived from the higher-level knapsack inequalities (for the shift loading capacities) over route variables.

Tilk et al. (2021) introduced the last-mile Vehicle Routing Problem with Delivery Options (VRPDO), in which some requests can be shipped to alternative locations with possibly different time windows. Moreover, when delivery options share a common location, e.g., a locker, capacities must be respected when assigning shipments. Thus, we have here the double knapsack structure, as customer deliveries are subject to both vehicle and delivery location capacities. Knapsack constraints are however not nested: two customer deliveries by the same vehicle do not necessarily contribute to the same higher-level knapsack constraints corresponding to the delivery location capacities. Tilk et al. (2021) propose a branch-cut-and-price algorithm for the VRPDO which is similar to the one for the standard VRPTW except for a different graph used when solving the pricing problem. No specific valid inequalities based on the knapsack structure of the problem are proposed.

Albareda-Sambola et al. (2009) introduced the capacity and distance-constrained plant location problem. It is an extension of the discrete capacitated plant location problem, where the customers assigned to each plant have to be packed in groups that will be served by one vehicle each. The constraints include two types of capacities. On the one hand, plants are capacitated, and the demands of the customers are indivisible. On the other hand, the total distance traveled by each vehicle to serve its assigned customers in round trips plant–customer–plant is also limited. This problem also has a nested knapsack structure. Here however different quantities contribute to the lower-level and higher-level knapsack constraints: plant–customer–plant distances to the former and customer demands to the latter. The authors proposed integer programming formulations and a tabu search heuristic. Later, Fazel-Zarandi and Beck (2012) proposed a logic-based Benders decomposition algorithm for this problem.

3 Formulation with a Polynomial Number of Variables and Additional Cuts

In this section, we present Formulation (F), the 3-index formulation proposed in Contardo et al. (2013). That formulation can be viewed as a disaggregated-by-depot version of the formulation introduced by Belenguer et al. (2011). Given sets $U, W \subseteq V$, let $\bar{U} = V \setminus U$ and denote by $\delta(U, W) \subseteq E$ the set of edges containing one endpoint in U and the other in W. We denote $\delta(U, \bar{U})$ simply by $\delta(U)$, and $\delta(\{v\}, V \setminus \{v\})$ by only $\delta(v)$. For a given set of customers $S \subseteq J$, let $d(S) = \sum_{j \in S} d_j$, and define $r(S) = \lceil d(S)/Q \rceil$ as a lower bound on the number of vehicles needed to serve all the customers in S.

3.1 Formulation (F)

For every depot location $i \in I$, define a binary variable y_i which takes value 1 if the depot i is opened, and zero otherwise. For every $i \in I$ and $j \in J$, define a binary variable z_{ij} that takes the value 1 if the customer j is assigned to depot i. For every depot $i \in I$ and customer $j \in J$, let $x^i_{(i,j)} \in \{0,1,2\}$ be a variable indicating how many times edge $(i,j) \in E_{IJ}$ is traversed by a route leaving depot i. If the customer j is served by a dedicated route incident to i, *i.e.*, a route of the form i-j-i, variable $x^i_{(i,j)}$ takes value 2. Finally, with every depot $i \in I$ and edge $(j,k) \in E_J$, let $x^i_{(j,k)}$ be a binary variable that takes 1 if a route incident to depot i traverses the edge (j,k), and zero, otherwise. The CLRP can be formulated as the following Mixed Integer Programming (MIP) problem:

$$(\mathrm{F}) \equiv \min \sum_{i \in I} f_i\, y_i + \sum_{i \in I} \sum_{e \in E} c_e\, x^i_e \tag{1}$$

$$\sum_{i \in I} z_{ij} = 1, \qquad \forall\, j \in J, \tag{2}$$

$$\sum_{e \in \delta(j)} x^i_e = 2\, z_{ij}, \qquad \forall\, i \in I,\ j \in J, \tag{3}$$

$$z_{ij} \le y_i, \qquad \forall\, i \in I,\ j \in J, \tag{4}$$

$$\sum_{j \in J} d_j\, z_{ij} \le W_i\, y_i, \qquad \forall\, i \in I, \tag{5}$$

$$\sum_{i \in I} \sum_{e \in \delta(S)} x^i_e \ge 2\, r(S), \qquad \forall\, S \subseteq J, \tag{6}$$

$$x^i_e \in \{0,1\}, \qquad \forall\, i \in I,\ e \in E_J, \tag{7}$$

$$x^i_e \in \{0,1,2\}, \qquad \forall\, i \in I,\ e \in E_{IJ}, \tag{8}$$

$$y_i \in \{0,1\}, \qquad \forall\, i \in I, \tag{9}$$

$$z_{ij} \in \{0,1\}, \qquad \forall\, i \in I,\ j \in J. \tag{10}$$

Inequalities (2) guarantee that every customer is assigned to exactly one depot. Inequalities (3) are the degree constraints for customer nodes. These inequalities assure that, if the customer j is served by depot i, then there must exist exactly two edges of a route leaving depot i which are incident to customer j. In the case that customer j is serviced by a dedicated route from depot i, this inequality is satisfied by variable x^i_{ij} assuming the value 2. Inequalities (4) have the format of *Generalized Upper Bound* (GUB) constraints and imply that a customer can only be served from an open depot. Inequalities (5) guarantee that the total demand supplied by the depot does not exceed its capacity. Inequalities (6), the so-called *Rounded Capacity Cuts* (RCC), introduced in the context of the CVRP (Laporte and Nobert, 1983), determine a lower bound on the minimum number of vehicles that must service S. Finally, inequalities (7)–(10) are the variable domains.

Even though inequalities (6) only express a lower bound on the number of vehicles needed to serve S, (F) is indeed a complete formulation for the CLRP, and not only a relaxation. This follows from the fact that every customer must be assigned to exactly one open depot, and once they are assigned (i.e., all y and z variables are fixed to 0 or 1), the problem decomposes into a number of independent CVRP-like subproblems, for which constraints (3), (6)-(8) suffice.

3.2 Additional Cuts

Since there are relatively many GUB inequalities (4), it is more efficient to separate only those that are violated. This can be easily done by inspection. There are an exponential number of RCC inequalities (6). They can be separated using the procedures described in Lysgaard et al.

(2004). In the remainder of this section, we present other inequalities used to strengthen (F).

3.2.1 Depot Cover Inequalities (COV)

We use cover inequalities to strengthen knapsack-like inequalities (5). The inequality corresponding to depot $i \in I$ differs from a standard knapsack inequality because its capacity W_i is multiplied by y_i. Yet, known inequalities for the standard knapsack problem can be readily adapted by also multiplying their right-hand side by y_i. Given a subset $J' \subset J$ of customers such that $\sum_{j \in J'} d_j > W_i$, the following *depot cover inequality*

$$\sum_{j \in J'} z_{ij} \leq (|J'| - 1)\, y_i \tag{11}$$

is valid for the CLRP .

These inequalities are exactly separated by a newly proposed procedure that generalizes the separation procedure for the standard knapsack cover inequality, as presented for instance in Wolsey (1998). Let $(\bar{y}, \bar{z})$ be a fractional solution to (F), restricted to variables y and z. The procedure consists in solving the following IP for each depot i such that $\bar{y}_i > 0$,

$$(\text{COV} - \text{Sep}) \equiv z = \min \sum_{j \in J} (\bar{y}_i - \bar{z}_{ij})\, w_j \tag{12}$$

$$\sum_{j \in J} d_j w_j \geq W_i + 1, \tag{13}$$

$$w_j \in \{0, 1\}, \qquad \forall\, j \in J. \tag{14}$$

Variable w_j equals 1 if customer j belongs to the cover J', 0 otherwise. Let z^* be the value of the optimal solution to the IP, if $z^* < \bar{y}_i$, then the depot cover inequality characterized by J' and i is violated by $\bar{y}_i - z^*$. The correctness of (COV-Sep) follows directly from the fact that (11) is equivalent to:

$$\sum_{j \in J'} (y_i - z_{ij}) \geq y_i. \tag{15}$$

By complementing the w variables, IP (COV-Sep) can be transformed into a 0-1 knapsack problem and solved by pseudo-polynomial algorithms that are very efficient in practice (Pferschy et al., 2004).

3.2.2 Fenchel Cuts (FC) over the y variables

In a feasible solution to the CLRP , the total capacity of the opened depots must be larger than the total demand of the customers: $\sum_{i \in I} W_i y_i \geq \sum_{j \in J} d_j$. We use a Fenchel Cut (FC) separation scheme to obtain new valid inequalities from that covering constraint. The separation of Fenchel cuts for MIPs was pioneered in Boyd (1993, 1994), and used, for example, by Boccia et al. (2008) to strengthen knapsack constraints. The idea of FC separation is to solve an LP in order to find the coefficients of a cutting plane separating the current fractional solution from the convex hull of the integer solutions of a certain subproblem (defined by a restricted set of variables and constraints).

Define the set of binary points satisfying the covering constraint as:

$$H = \left\{ h \mid \sum_{i \in I} W_i h_i \geq \sum_{j \in J} d_j \;, h \in \{0,1\}^{|I|} \right\}. \tag{16}$$

Let $\bar{y}$ be the current fractional solution of (F) restricted to y and define a vector of variables α

having dimension $|I|$.

$$(\text{FC} - \text{Sep}) \equiv z = \min \sum_{i \in I} \bar{y}_i \, \alpha_i \tag{17}$$

$$\sum_{i \in I} h_i \, \alpha_i \geq 1, \quad \forall \, h \in H, \tag{18}$$

$$\alpha_i \geq 0, \quad \forall \, i \in I. \tag{19}$$

After solving that LP, if $z^* < 1$, then $\sum_{i \in I} \alpha_i \, y_i \geq 1$ is a valid Fenchel cut that is violated by $\bar{y}$.

The potential difficulty with that Fenchel cut separation is the enumeration of the set H. A point $h \in H$ is said to be minimal if there is no other point $h' \in H$ such that $h' \leq h$. It can be seen that non-minimal points lead to redundant constraints in (18). So, it suffices to enumerate the set of minimal points in H. In our experiments with CLRP, as all instances from the literature have $|I| \leq 20$, solving (FC-Sep) was never too time-consuming. However, for larger values of $|I|$ it may be necessary to generate constraints (18) dynamically, as done in Boccia et al. (2008).

3.2.3 Depot Capacity Cuts (DCC)

A generalization of rounded capacity cuts is the so-called *depot capacity cuts* introduced by Belenguer et al. (2011). Let $R \subseteq I$ and $S \subseteq J$ be such that $d(S) > W(R)$, implying that the available capacity of depots in R is not sufficient to meet the demand of S. Let $r(S, R) = \lceil (d(S) - W(R))/Q \rceil$ be a lower bound on the number of vehicles coming from $I \setminus R$ needed to serve set S. Then, a *depot capacity cut* is the following inequality:

$$\sum_{i \in I \setminus R} \sum_{e \in \delta(S)} x_e^i \geq 2r(S, \, R). \tag{20}$$

It can be seen that Rounded Capacity Cuts (6) correspond to the depot capacity cuts where $R = \emptyset$.

Belenguer et al. (2011) have also verified that it is possible to strengthen inequalities (20), in what they called *improved depot capacity cuts*. The idea is to consider the effect of variable y_{i_1}, for some depot $i_1 \in R$. The whole family of cuts can be written as:

$$\begin{gathered} \sum_{i \in I \setminus R} \sum_{e \in \delta(S)} x_e^i \geq 2r(S, R) + 2(1 - y_{i_1}) \big(r(S, R \setminus \{i_1\}) - r(S, R) \big), \\ \forall \, R \subseteq I, \, i_1 \in R, \, S \subseteq J : d(S) > W(R). \end{gathered} \tag{21}$$

The validity of improved depot capacity cuts can be proved by considering the two possible values for y_{i_1}. When $y_{i_1} = 1$, inequality (21) reduces to (20). On the other hand, for the case where $y_{i_1} = 0$, the right-hand side of inequality (21) becomes $r(S, R \setminus \{i_1\})$, which is correct. An improved depot capacity cut clearly dominates the corresponding inequality (20) when $r(S, R \setminus \{i_1\}) > r(S, R)$. Contardo et al. (2013) proposed alternative inequalities that also strengthen inequalities (20) by considering the y variable of a single depot in R.

By extending those lines of reasoning to two depots in R, we now propose a new family of improved depot capacity cuts.

Theorem 1. *Given $S \subseteq J$, and $R \subseteq I$ such that $d(S) > W(R)$, and a pair of vertices $i_1, i_2 \in R$, the following inequality is valid for formulation (F)*

$$\begin{aligned} \textstyle\sum_{i \in I \setminus R} \sum_{e \in \delta(S)} x_e^i \geq \; & 2r(S, R) y_{i_1} + 2r(S, R \setminus \{i_1\}) y_{i_2} \\ & + 2r(S, R \setminus \{i_1, i_2\})(1 - y_{i_1} - y_{i_2}). \end{aligned} \tag{22}$$

Proof. First consider the case where $y_{i_1} = y_{i_2} = 1$. The right-hand side of (22) simplifies to $2r(S, R) + \big(2r(S, R \setminus \{i_1\}) - 2r(S, R \setminus \{i_1, i_2\}) \big)$. Since $2r(S, R \setminus \{i_1\}) - 2r(S, R \setminus \{i_1, i_2\}) \leq 0$,

the inequality is redundant with respect to inequality (20). Then, consider a solution for which $y_{i_1} = 1$ and $y_{i_2} = 0$. In this case, the right-hand side of (22) reduces to $2r(S, R)$, and this inequality is equivalent to (20). If $y_{i_1} = 0$ and $y_{i_2} = 1$, the right-hand side of (22) becomes $2r(S, R \setminus \{i_2\})$, which is correct. Finally, if $y_{i_1} = y_{i_2} = 0$ then the right-hand side of (22) becomes $2r(S, R \setminus \{i_1, i_2\})$, which is also correct. □

In what follows, we describe our heuristic separation for inequalities (21) and (22). Let $(\bar{x}, \bar{y})$ be a fractional solution to (F), restricted to variables x and y. If the problem variant is solved in which all depots are already opened, we set $\bar{y}_i = 1$ for all $i \in I$. The heuristic considers the collection $\mathcal{R}$ of all depot subsets R such that $1 \leq |R| \leq 2$ and $\bar{y}_i > 0$ for all $i \in R$. This collection is randomly shuffled, and first $\lceil \psi \cdot |I| \rceil$ subsets are selected. We use values $\psi = 7$ if $|I| < 15$, $\psi = 5$ if $15 \leq |I| < 20$, and $\psi = 3.6$ if $20 \leq |I|$. The same seed is always used to initialize the random number generator in order to have the deterministic behavior of the algorithm.

On every iteration, we fix a depot subset R in the collection of selected subsets. Let $\tilde{S}(R)$ be the set of customers who are visited only by vehicles starting in depots in R: $\tilde{S}(R) = \{j \in J : \sum_{i \in R} \sum_{e \in \delta(j)} \bar{x}_e^i = 2\}$. Then we use the following greedy construction heuristic similar to the one presented by Lysgaard et al. (2004) for separating RCCs. For each seed customer $j \in J \setminus \tilde{S}(R)$, we set $S = \tilde{S}(R) \cup \{j\}$ and then iteratively expand S by one customer and check inequalities (21) and (22) for the resulting pair (R, S) and all possible $i_1, i_2 \in R$. If all depots are already open, then only the original inequality (20) is verified for violation. The customer j we add to S is the one that minimizes the value $\sum_{i \in I \setminus R} \sum_{e \in \delta(S \cup \{j\})} \bar{x}_e^i$ subject to the restriction that we have not generated the set $S \cup \{j\}$ before during the current iteration with fixed R. When we cannot expand the current set S without generating a previously generated set, we proceed to the next seed.

4 Extended Formulation with an Exponential Number of Variables

4.1 Formulation (EF)

The linear relaxation of formulation (F), even with the previously presented valid inequalities, provides lower bounds that are not strong enough for developing a state-of-the-art algorithm. Hence, we present a stronger extended formulation obtained by introducing route variables. Given $i \in I$, we denote by $\Omega(i)$ the set of all possible routes incident to depot i. The set of all routes, i.e., $\cup_{i \in I} \Omega(i)$, will be denoted simply by Ω. Furthermore, given $\omega \in \Omega$, $i(\omega) \in I$ denotes the depot in which route ω is incident to, $J(\omega) \subseteq J$ is the set of customers visited by route ω. For each route $\omega \in \Omega$, we associate a *load*, denoted by $d(\omega)$, which is given by the sum of demands of the customers visited by the route, *i.e.*, $d(\omega) = \sum_{j \in J(\omega)} d_j$. Moreover, for each route $\omega \in \Omega$, we define the *cost* $c(\omega)$ as the sum of the costs of the edges traversed by the route, *i.e.*, $c(\omega) = \sum_{e \in E} b_e^\omega c_e$, where b_e^ω is the number of times edge e appears in ω (this coefficient can have values 0, 1 or 2; the last case happens when ω visits a single customer j, so edge $(i(\omega), j)$ is used twice). Given $\omega \in \Omega$, let a_j^ω be a binary coefficient indicating whether customer $j \in J$ is visited by route ω. For every route $\omega \in \Omega$, let variable λ_ω indicate whether route ω is used in the solution. New variables λ can be linked to the x and z variables in formulation (F) by the following equations:

$$x_e^i = \sum_{\omega \in \Omega(i)} b_e^\omega \lambda_\omega, \quad \forall\, i \in I,\, e \in E, \tag{23}$$

$$z_{ij} = \sum_{\omega \in \Omega(i)} a_j^\omega \lambda_\omega, \quad \forall\, i \in I,\, j \in J. \tag{24}$$

Performing those variable substitutions, the following Extended Formulation is obtained:

$$(\text{EF}) \equiv \min \sum_{i \in I} f_i \, y_i + \sum_{\omega \in \Omega} \left(\sum_{e \in E} b_e^\omega \, c_e \right) \lambda_\omega \tag{25}$$

$$\sum_{\omega \in \Omega} a_j^\omega \, \lambda_\omega = 1, \qquad \forall \, j \in J, \tag{26}$$

$$\sum_{\omega \in \Omega(i)} a_j^\omega \, \lambda_\omega \leq y_j, \qquad \forall \, i \in I, \; j \in J, \tag{27}$$

$$\sum_{\omega \in \Omega(i)} \left(\sum_{j \in J} d_j a_j^\omega \right) \lambda_\omega \leq W_i \, y_i, \qquad \forall \, i \in I, \tag{28}$$

$$\sum_{\omega \in \Omega} \left(\sum_{e \in \delta(S)} b_e^\omega \right) \lambda_\omega \geq 2 \, r(S), \qquad \forall \, S \subseteq J, \tag{29}$$

$$\sum_{\omega \in \Omega(i)} b_e^\omega \, \lambda_\omega \in \{0, 1\}, \qquad \forall \, i \in I, \; e \in E_J, \tag{30}$$

$$\sum_{\omega \in \Omega(i)} b_e^\omega \, \lambda_\omega \in \{0, 1, 2\}, \quad \forall \, i \in I, \; e \in E_{IJ}, \tag{31}$$

$$y_i \in \{0, 1\}, \qquad \forall \, i \in I, \tag{32}$$

$$\sum_{\omega \in \Omega(i)} a_j^\omega \, \lambda_\omega \in \{0, 1\}, \qquad \forall \, i \in I, \; j \in J, \tag{33}$$

$$\lambda_\omega \in \{0, 1\} \qquad \forall \, \omega \in \Omega. \tag{34}$$

The objective function and the constraints in (25)–(33) correspond directly to (1)–(10), except that there is no counterpart for equalities (3). Those equalities, that would be translated as

$$\sum_{\omega \in \Omega(i)} \left(\sum_{e \in \delta(j)} b_e^\omega \right) \lambda_\omega = \sum_{\omega \in \Omega(i)} 2 a_j^\omega \, \lambda_\omega, \quad \forall \, i \in I, \, j \in J, \tag{35}$$

would be redundant in (EF), since the definition of route implies that $\sum_{e \in \delta(j)} b_e^\omega = 2 a_j^\omega$ for every $j \in J$ and $\omega \in \Omega$. Inequalities (27) are still called GUBs, while (29) are still RCCs. Constraints (30) and (31) are equivalent to the integrality constraints over the x variables. Constraints (33) are equivalent to the integrality constraints over the z variables.

When solving the linear relaxation of (EF) by column generation, both GUBs and RCCs should still be separated on demand. Other inequalities defined over the x, y and z variables, like COVs, FCs, and DCCs, can also be added to (EF), after their x and z terms are translated to λ using expressions (23) and (24), respectively. We remark that it is never necessary to branch over individual λ variables (which would make the pricing subproblem harder) in order to have an integer solution. Since formulation (EF) is a stronger extension of a complete formulation (F), it is enough to make sure that y, x and z are integer.

4.2 Limited Memory Rank-1 Cuts

Rank-1 cuts for vehicle routing problems are obtained by the Chvátal-Gomory rounding of set-partitioning constraints (26) relaxed to ≤ 1 inequalities. For a non-negative vector $\alpha \in \mathbb{Z}_+^{|J|}$ of multipliers, the following rank-1 cut is valid:

$$\sum_{\omega \in \Omega} \left\lfloor \sum_{j \in J} \alpha_j a_j^w \right\rfloor \lambda_\omega \leq \left\lfloor \sum_{j \in J} \alpha_j \right\rfloor . \tag{36}$$

An inequality (36) obtained using a vector of multipliers with k positive components is called a k-row rank-1 cut. If all positive components of α are the same, the corresponding inequality is called a subset-row cut. Jepsen et al. (2008) introduced 3-row and 5-row subset-row cuts. Petersen et al. (2008) first considered using general rank-1 cuts of format (36). Pecin et al. (2017b) studied k-row rank-1 cuts with $k \leq 5$, determining all dominant vectors of multipliers: if a k-row rank-1 cut with $k \leq 5$ is violated, then at least one rank-1 cut obtained using a dominant vector of multipliers is also violated. Those are exactly the multipliers used in this work.

The use of the limited memory variant of those cuts, introduced by Pecin et al. (2017a), is crucial to reduce the impact of those non-robust cuts on the practical performance of the labeling algorithm used in the pricing.

4.3 Branch-Cut-and-Price algorithm

Formulation (EF) plus robust cuts, together with limited-memory rank-1 cuts, can be solved by an adaptation of the Branch-and-Cut-and-Price (BCP) algorithm presented in Sadykov et al. (2021) for the Heterogeneous Fleet Vehicle Routing Problem (HFVRP). The adaptation is relatively straightforward because each of the multiple depots can be viewed as defining a different vehicle type.

In that BCP algorithm, the linear relaxation of (EF), i.e., the master problem, is solved by a column and cut generation procedure. Column generation is an iterative approach that alternates between solving the master problem with a restricted number of variables λ and the pricing problem. The latter is solved to find variables λ with a negative reduced cost if they exist, given the current dual solution of the restricted master. The pricing problem, decomposed into subproblems by vehicle types (depots), is solved by the bucket-graph based labeling algorithm. For the LRP and VRP-CMD that algorithm considers a single resource, corresponding to the vehicle capacity. For VRPTW-S there is an additional resource for considering the time windows. In addition, the bucket arc elimination procedure is employed to remove arcs from the bucket graph used by the labeling algorithm and thus, reduce its running time in future iterations. An elementary route enumeration procedure is also exploited to enumerate all routes which can participate in an improving solution. If the enumeration is successful for a vehicle type, the corresponding pricing subproblem is solved by inspection. If all pricing subproblems are enumerated, and the total number of enumeration routes is small, the formulation (EF) with all corresponding variables λ is solved by the MIP solver.

The y variables, some constraints and also the robust cuts (described in Section 3.2) used in formulation (EF) do not exist in the HFVRP formulation. However, they only modify the master problem and do not have any impact on the structure of the pricing problem. So, the bucket graph based labeling algorithm used to solve it does not need to change.

As in (Sadykov et al., 2021), branching is performed by adding constraints to the master LP that correspond to tightening lower and upper bounds on the values of some aggregated variables. The following aggregated variables are used for this purpose.

- The number of open depots in a subset $I' \subseteq I$, $1 \leq |I'| \leq 4$: $g^{\mathrm{ND}}_{I'} = \sum_{i \in I'} y_i$.
- The number of vehicles starting in depot $i \in I$: $g^{\mathrm{VN}}_i = \frac{1}{2} \sum_{j \in J} x^i_{ij}$.
- The total number of vehicles: $g^{\mathrm{TVN}} = \sum_{i \in I} g^{\mathrm{VN}}_i$.
- The number of customers served from depot $i \in I$: $g^{\mathrm{CN}}_i = \sum_{j \in J} z_{ij}$.
- Assignment of customer $j \in J$ to depot $i \in I$: z_{ij}.
- Participation of edge $e \in E$ in the solution: $g^{\mathrm{EDG}}_e = \sum_{i \in I} x^i_e$.

Branching on the number of open depots has a higher priority. Branching on other aggregated variables has the same but lower priority. The branching aggregated variable is chosen among

ones with the same priority by the multi-phase strong branching procedure, described in (Sadykov et al., 2021).

To improve feasible solutions, after each node in the branch-and-bound tree we use the following heuristic similar to the one introduced by Pessoa et al. (2009). In an iterative procedure, we decrease the artificial bound in order to divide the primal-dual gap by two in each iteration. Then, we perform elementary route enumeration for each pricing subproblem. The iterative procedure stops when the enumeration succeeds for all subproblems. Afterward, we pick 5000 elementary routes with the smallest reduced cost and add them to the master problem. Finally, we use IBM CPLEX MIP to solve the resulting problem with a time limit of 20 seconds. We activate the polishing heuristic (Rothberg, 2007) implemented in CPLEX.

5 New Family of Non-Robust Cuts

In this section, we present a new family of valid inequalities derived from knapsack-like constraints (28). These inequalities are non-robust and do change the pricing. We make extensive use of the properties that characterize the facets of the so-called master knapsack polytope, *properties that are used in the definition of the newly proposed cuts*, in order to propose effective separation procedures and also for minimizing the necessary modifications to the labeling algorithm using in the pricing, keeping it efficient even after an arbitrary number of such cuts are added.

5.1 Route Load Knapsack Cuts

The *master knapsack polytope* is defined as:

$$\mathcal{P}_{\mathrm{MKP}}(W) = \mathrm{conv}\left\{ t \in \mathbb{Z}_+^W : \sum_{q=1}^{W} q\, t_q \leq W \right\}.$$

Next theorem characterizes the non-trivial facets (those that are not described by non-negativity inequalities $t_q \geq 0$, $1 \leq q \leq W$, or by $\sum_{q=1}^{W} q\, t_q \leq W$ itself) of this polytope.

Theorem 2 (Aráoz (1974), Aráoz et al. (2003))**.** *Each non-trivial facet of $\mathcal{P}_{MKP}(W)$ can be described by an inequality of format $\xi t \leq 1$ such that the coefficient vector $\xi \in \mathbb{R}_+^W$ is an extreme point of the following system of linear constraints:*

$$\xi_1 = 0, \tag{37}$$

$$\xi_W = 1, \tag{38}$$

$$\xi_q + \xi_{W-q} = 1, \qquad \forall 1 \leq q \leq W/2, \tag{39}$$

$$\xi_q + \xi_{q'} \leq \xi_{q+q'}, \quad \forall q + q' \leq W. \tag{40}$$

Constraints (39) and (40), define, respectively, the *complementarity* and *superadditivity* properties that should be satisfied by any inequality defining a non-trivial facet $\mathcal{P}_{\mathrm{MKP}}(W)$. The superadditivity together with (37) implies the *monotonicity* property (the coefficients in vector ξ are non-decreasing) because $\xi_q = 0 + \xi_q = \xi_1 + \xi_q \leq \xi_{q+1}$.

The master polytope $\mathcal{P}_{\mathrm{MKP}}(W)$ corresponds to an integer knapsack problem with right-hand side W where all possible left-hand side coefficients, from 1 to W, do exist. General integer knapsack problems do not have all possible left-hand side coefficients. However, all non-trivial facets of a general integer knapsack polytope with right-hand side W are projections (obtained by eliminating the non-existing coefficients) of some facet of $\mathcal{P}_{\mathrm{MKP}}(W)$ (see Chopra et al. (2015)). Therefore, Theorem 2 also yields a (pseudo-polynomial) exact LP-based separation procedure for the general integer knapsack polytope. For example, consider the general integer knapsack problem with three variables defined by $2t_2 + 3t_3 + 4t_4 \leq 5$, $t \in \mathbb{Z}_+^3$. A solution to its linear

relaxation is $t_2 = 2$, $t_3 = 1/3$, and $t_4 = 0$. We can cut that point by solving the following LP: $z = \max 2\xi_2 + 1/3\,\xi_3$ subject to $\xi_1 = 0$, $\xi_5 = 1$, $\xi_1 + \xi_4 = 1$, $\xi_2 + \xi_3 = 1$, $2\xi_1 + \leq \xi_2$, $\xi_1 + \xi_2 \leq \xi_3$, $\xi_1 + \xi_3 \leq \xi_4$, $\xi_1 + \xi_4 \leq \xi_5$, $2\xi_2 + \leq \xi_4$, $\xi_2 + \xi_3 \leq \xi_5$. The optimal solution to that LP is the extreme point $\xi_1 = 0$, $\xi_2 = \xi_3 = 0.5$, $\xi_3 = \xi_4 = 1$, with $z = 7/6$. So, by Theorem 2, $0.5t_2 + 0.5t_3 + t_4 + t_5 \leq 1$ defines a facet of $\mathcal{P}_{\text{MKP}}(5)$. By eliminating the non-existing coefficients one obtains $0.5t_2 + 0.5t_3 + t_4 \leq 1$, a facet-defining cut for the original knapsack problem. This LP-based separation procedure is rarely used in practice for the following reason: even if the integer knapsack problem is very sparse, having only a few non-zero coefficients, it is still necessary to solve an LP with W variables and $O(W^2)$ constraints. Yet, the theoretical properties characterized in Theorem 2 may be very helpful, as will be seen next.

Define θ^i_q as an integer variable indicating how many routes with a total load of *exactly* q units, where $1 \leq q \leq W_i$, leave depot $i \in I$. Those variables θ are simply an aggregation of variables λ and can be expressed as:

$$\theta^i_q = \sum_{\omega \in \Omega(i):\, d(\omega) = q} \lambda_\omega, \quad \forall\, i \in I,\ 1 \leq q \leq W_i. \tag{41}$$

Then, inequalities (28) imply that:

$$\sum_{q=1}^{W_i} q\, \theta^i_q \leq W_i\, y_i, \quad i \in I. \tag{42}$$

We want to obtain strong cuts from (42). The following result proved in the Appendix, shows that Theorem 2 still provides a characterization of the desired inequalities.

Theorem 3. *$\xi t \leq 1$ defines a non-trivial facet of $\mathcal{P}_{MKP}(W_i)$ if and only if $\xi\theta \leq y_i$ defines a non-trivial facet of* $\text{conv}\{(\theta^i, y_i) \in \mathbb{Z}^{W_i}_+ \times \{0,1\} :\ \sum_{q=1}^{W_i} q\,\theta_q \leq W_i y_i\}$.

However, in order to also obtain some inequalities that are not facet-defining but can be cheaply separated, we exclude the complementarity conditions (39) from the following definition.

Definition 1. *Given a depot $i \in I$ and a vector $\xi \in \mathbb{R}^{W_i}_+$ satisfying* (37)–(38) *and* (40)*, the inequality*

$$\sum_{q=1}^{W_i} \xi_q\, \theta^i_q \leq y_i, \tag{43}$$

is known as a Route Load Knapsack Cut *(RLKC).*

Theorem 4. *A Route Load Knapsack Cut* (43) *is valid for (EF).*

Proof. Let $(\bar{\theta^i}, \bar{y}_i) \in \mathbb{Z}^{W_i}_+ \times \{0,1\}$ be an integer solution of (EF) restricted to the variables θ and y related to depot i. If $\bar{y}_i = 0$, then $\bar{\theta^i}_q = 0$ for all $1 \leq q \leq W_i$, due to (42). Thus, inequality (43) is satisfied.

Consider now the case $\bar{y}_i = 1$. Let $\xi(q) = \xi_q$ be a function defined for integer values between 1 and W_i. By the repeated application of (37)–(38), (40) and due to the fact that $\bar{\theta^i}$ satisfies (42), we have that

$$\sum_{q=1}^{W_i} \xi_q \bar{\theta}^i_q \leq \xi(\sum_{q=1}^{W_i} q\bar{\theta}^i_q) \leq \xi(W_i) = 1 = \bar{y}_i.$$

Thus, inequality (43) is also satisfied. □

In order to illustrate the proof for the case $\bar{y}_i = 1$, consider an example where $W_i = 6$ and $\bar{\theta^i} = (1, 2, 0, 0, 0, 0)$. Then, $\xi_1 + \xi_2 + \xi_2 \leq \xi_3 + \xi_2 \leq \xi_5 = \xi(5) \leq \xi(6) = 1 = \bar{y}_i$.

A Chvàtal-Gomory rounding of constraints (42) using multipliers $\beta \in \mathbb{R}$, where $\beta \geq 1/W_i$, obtain inequalities

$$\sum_{q=1}^{W_i} \frac{\lfloor \beta\, q \rfloor}{\lfloor \beta\, W_i \rfloor} \theta_q^i \leq y_i, \quad i \in I, \tag{44}$$

which are RLKCs. The supperadditive property of the coefficients follows from the fact that $\lfloor r \rfloor + \lfloor r' \rfloor \leq \lfloor r + r' \rfloor$ for all $r, r' \in \mathbb{R}_+$. However, the complementarity property is usually not satisfied.

5.2 Separation

Separation of RLKCs is done separately for each depot $i \in I$. For clarity, we omit index i for the remainder of this section: we have $\Omega = \Omega(i)$, $\theta = \theta^i$, $y = y_i$, $W = W_i$. Let $(\bar{\theta}, \bar{y})$ denote the current fractional solution.

First, we separate RLKCs by Chvàtal-Gomory rounding. For every q such $\bar{\theta}_q > 0$, we consider multipliers $\beta = p/q$, $p = 1, \ldots, q-1$, and check whether constraint (44) is violated.

Secondly, we separate stronger RLKCs, where the ξ coefficients satisfy all the conditions in Theorem 2. It would be possible to perform an exact separation by solving: maximize $z = \sum_{i=1}^{W} \bar{\theta}_q \xi_q$, subject to (37)–(40). If $z > y$, then inequality (43) would be violated. Moreover, by solving that LP using a simplex-based algorithm, the obtained coefficients would be extreme points of (37)–(40), so the separated RLKCs would always be facet-defining. However, as there are W variables and $O(W^2)$ constraints in that LP, that would be too time-consuming for large values of W. Also, the generation of a single violated cut per iteration would be undesirable due to possible convergence issues.

Therefore, we separate the RLKCs that correspond to the $1/k$-facets of the master knapsack polytope. A non-trivial facet $\xi t \leq 1$ of $\mathcal{P}_{\text{MKP}}(W)$ is called an *$1/k$-facet* if k is the smallest possible integer such that

$$\xi_q \in \{0/k,\, 1/k,\, 2/k, \ldots, k/k\} \cup 1/2, \quad 1 \leq q \leq W. \tag{45}$$

Using both theoretical arguments and computational experiments, it was shown in Chopra et al. (2015) that the $1/k$-facets for small values of k are by far the most important facets for obtaining a good approximation to $\mathcal{P}_{\text{MKP}}(W)$.

Let $\xi \in \mathbb{R}_+^W$ be a non-decreasing vector that satisfies (37)–(38) and (45), for a certain k. Such a vector can be uniquely determined by a non-decreasing sequence (a_m) where a_m represents the first index q with $\xi_q \geq m/k$ for $m \in \{1, \ldots, k\} \cup \{k/2\}$ (a_0 is not part of the sequence because it would always have value 1). The vector ξ corresponding to a certain value of k and to (a_m) will be denoted by $\xi^{k-(a_m)}$.

Theorem 5 (Chopra et al. (2015))**.** *A vector $\xi^{k-(a_m)}$ satisfies* (37)–(40) *if and only if*

$$2 \leq a_m \leq a_{m'} \leq (W+1)/2, \quad \forall m < m' \leq k/2, \tag{46}$$

$$a_m + a_{k+1-\lceil m \rceil} = W + 1, \quad \forall m \leq k/2, \tag{47}$$

$$a_m + a_{m'} \geq a_{\lceil m+m' \rceil} \quad \forall m \leq m' \text{ with } \lceil m + m' \rceil \leq k. \tag{48}$$

In that case, $\xi^{k-(a_m)}$ is said to define a $1/k$-inequality

In our separation algorithm, we enumerate non-decreasing sequences (a_m) satisfying constraints (46)–(48) that correspond to 1/6-, 1/8-, or 1/10-inequalities. Let $\Phi = \{q : 1 \leq q \leq W,\, \bar{\theta}_q > 0\}$.

To separate 1/6-inequalities $\xi^{6-(a_m)}$, we enumerate triples (a_1, a_2, a_3) such that $a_m \in \Phi$ or $\{W - a_m\} \in \Phi$ for $m = 1, \ldots 3$, and $2 \leq a_1 \leq a_2 \leq a_3 \leq (W+1)/2$. Values a_4, a_5, a_6 are then obtained from equalities (47). According to constraints (48), for each triple (a_1, a_2, a_3), we verify

$2a_1 \geq a_2$, $a_1 + a_2 \geq a_3$, $a_1 + 2a_3 \geq W + 1$ (obtained from $a_1 + a_3 \geq a_4$ or from $a_3 + a_3 \geq a_6$), and $2a_2 + a_3 \geq W + 1$ (obtained from $a_2 + a_3 \geq a_5$ or from $a_2 + a_2 \geq a_4$).

As an example, consider a case where $Q = 70$, $W = 140$ and a fractional solution where the non-zeros are $\bar{\theta}_{38} = 1/14$, $\bar{\theta}_{53} = 1/2$, $\bar{\theta}_{65} = 16/14$, $\bar{\theta}_{70} = 1/2$, and $\bar{y} = 277/280$. The best rounding cut, obtained with multiplier $\beta = 1/53$, is $\sum_{q=53}^{105} \frac{1}{2}\theta_q + \sum_{q=106}^{140} \theta_q \leq y$, yielding a violation of around 0.082. That cut is not facet-defining. However, there is a better 1/6-inequality cutting that fractional point. By taking $a_1 = 38$ and $a_2 = a_3 = 53$ (so, $a_4 = a_5 = 88$ and $a_6 = 103$), we get the facet-defining $\sum_{q=38}^{52} \frac{1}{6}\theta_q + \sum_{q=53}^{87} \frac{1}{2}\theta_q + \sum_{q=88}^{102} \frac{5}{6}\theta_q + \sum_{q=103}^{140} \theta_q \leq y$, yielding a violation of around 0.094.

To separate 1/8-inequalities $\xi^{8-(a_m)}$, we enumerate tuples (a_1, a_2, a_3, a_4) such that $a_m \in \Phi$ or $\{W - a_m\} \in \Phi$ for $m = 1, \ldots 4$, and $2 \leq a_1 \leq a_2 \leq a_3 \leq a_4 \leq (W + 1)/2$. Values a_5, a_6, a_7, a_8 are then obtained from equalities (47). According to constraints (48), for each tuple (a_1, a_2, a_3, a_4) we verify $2a_1 \geq a_2$, $a_1 + a_2 \geq a_3$, $a_1 + a_3 \geq a_4$, $2a_2 \geq a_4$, $a_1 + 2a_4 \geq W + 1$, $a_2 + a_3 + a_4 \geq W + 1$, and $3a_3 \geq W + 1$.

To separate 1/10-inequalities $\xi^{10-(a_m)}$, we enumerate tuples $(a_1, a_2, a_3, a_4, a_5)$ such that $a_m \in \Phi$ or $\{W - a_m\} \in \Phi$ for $m = 1, \ldots 5$, and $2 \leq a_1 \leq a_2 \leq a_3 \leq a_4 \leq a_5 \leq (W + 1)/2$. Values a_6, a_7, a_8, a_9, a_{10} are then obtained from equalities (47). According to constraints (48), for each tuple $(a_1, a_2, a_3, a_4, a_5)$ we verify $2a_1 \geq a_2$, $a_1 + a_2 \geq a_3$, $a_1 + a_3 \geq a_4$, $a_1 + a_4 \geq a_5$, $2a_2 \geq a_4$, $a_2 + a_3 \geq a_5$, $a_1 + 2a_5 \geq W + 1$, $a_2 + a_4 + a_5 \geq W + 1$, $a_3 + 2a_4 \geq W + 1$, and $2a_3 + a_5 \geq W + 1$.

For each sequence (a_m) which verifies (46)–(48), we generate vector $\xi^{k-(a_m)}$, and check if the corresponding inequality (43) is violated by $\bar{\theta}$. If a positive violation is found, the inequality is added to the restricted master problem. Some additional remarks on this separation procedure:

- A separated $1/k$- inequality, defined by $\xi^{k-(a_m)}$, is not necessarily a facet (it is if and only if that vector is an extreme point of (37)–(40)). Yet, repeated separation rounds until no violation is found obtains exactly the same bounds that would be obtained by only separating $1/k$-facets.
- The enumerative approach used in the procedure is likely to be effective even for large values of W because of the typical sparsity of the fractional solution vector (if W is large, usually $|\Phi| << W$).
- An $1/k'$-inequality is also an $1/k$-inequality if k' is a divisor of k. So, 1/2-, 1/3-, 1/4-, and 1/5-inequalities are also being separated by the procedure.

5.3 Modifying the Labeling Algorithm

Given a dual solution to the linear relaxation of the restricted Master Formulation, the pricing problem searches for variables λ_ω with a negative reduced cost. The pricing problem can be decomposed into $|I|$ similar subproblems, one for each depot $i \in I$. In this section, we consider a pricing subproblem for a fixed depot i.

In the absence of non-robust Rank-1 cuts (36) and RLKCs (43), the reduced cost of variable λ_ω corresponding to route $\omega \in \Omega(i)$ can be expressed as the sum of reduced costs for every edge in the route, defined as follows. Assume that there are M active robust constraints or cuts, the m-th such constraint having dual variable τ_m. Using the fact that $z_{ij} = 1/2 \sum_{e \in \delta(j)} x^i_e$, $i \in I$, $j \in J$ (Equations (3)), we can eliminate the z variables and express the robust constraint m in format:

$$\sum_{i \in I} \sum_{e \in E} \eta^{im}_e x^i_e + \sum_{i \in I} \iota^m_i y_i \leq b_m.$$

Actually, Constraints (2) are equalities, but this does not make difference. So, the reduced cost of an edge e for a subproblem $i \in I$ is given by:

$$\bar{c}^i_e = c_e - \sum_{m=1}^{M} \eta^{im}_e \tau_m.$$

Then, the pricing problem is modeled as a resource-constrained elementary shortest path problem (RCSPP) in the following complete directed graph $D^i = (\{i\} \cup J, A)$. The reduced cost $\bar{c}_a$ of every pair of opposite arcs (u, v) and $(v, u) \in A$, corresponding to $e = (u, v) \in E$, is set to $\bar{c}^i_e$. We have a single capacity resource. The resource consumption of arc $(j, j') \in A$ is equal to $\frac{1}{2}d_j + \frac{1}{2}d_{j'}$, considering that $d_i = 0$. That symmetric definition, in the sense that the consumption of arcs (j, j') and (j', j) are always identical, improves the efficiency of the labeling algorithm. Bounds on the accumulated resource consumption are $[0, Q]$, for every node $j \in \{i\} \cup J$. The RCSPP is then to find an elementary cycle of minimum reduced cost starting and finishing in node i. We apply the bi-directional bucket graph based labeling algorithm proposed by Sadykov et al. (2021) to solve this RCSPP. That labeling algorithm already incorporates the modifications needed for handling limited-memory Rank-1 cuts and the *ng*-route relaxation, following ideas from Baldacci et al. (2011a); Jepsen et al. (2008); Pecin et al. (2017a).

The main goal of this section is to describe the additional modifications of the basic labeling algorithm for handling RLKPs (43). Assume that there is a set Γ of active RLKPs, constraint $\gamma \in \Gamma$ having negative dual variable π_γ. The coefficient of a variable λ_ω in an RLKC γ is defined by a non-decreasing and superadditive function $\xi_\gamma(d(\omega))$ on the route total load $d(\omega)$. Thus, the total contribution of all RLKCs to the reduced cost of λ_ω is $g(d(\omega)) = \sum_{\gamma \in \Gamma} -\pi_\gamma \xi_\gamma(d(\omega))$. Being a positive linear combination of functions $\xi_\gamma(d(\omega))$, function $g(d(\omega))$ is also non-decreasing and superadditive.

In the labeling algorithm, each label L represents a partial path $\omega(L)$ from the node i. Let $J(L)$ be the set of customers visited by the partial path, $j(L)$ be the final node of the partial path, $\bar{c}(L)$ be the sum of reduced costs of arcs in the partial path, and $q(L)$ be the total capacity resource consumption of the partial path. The algorithm consists in an enumeration of all feasible partial paths. For that, every label L is extended by taking each arc $a' = (j(L), j')$ outgoing from node $j(L)$. After extension, a new label L' is created, for which, $j(L') = j'$, $q(L') = q(L) + \frac{1}{2}d_{j(L)} + \frac{1}{2}d_{j(L')}$, and $\bar{c}(L') = \bar{c}(L) + \bar{c}_{a'}$. To avoid complete enumeration, a dominance rule should be used to remove labels that cannot lead to the minimum reduced cost path when extended. A label L dominates label L' if for any completion path ω such that union of paths $\omega(L')$ and ω is feasible, i.e., $(\omega(L'), \omega) \in \Omega$, we have i) union of paths $\omega(L)$ and ω is also feasible and ii) reduced cost of path $(\omega(L'), \omega)$ is not smaller than the reduced cost of path $(\omega(L), \omega)$. The next theorem gives a valid dominance rule.

Theorem 6. *Given a non-decreasing function $g(d(\omega))$ representing the contribution of the RLKCs to the reduced cost of a path $\omega \in \Omega(i)$, label L dominates label L' if $j(L) = j(L')$, $J(L) \subseteq J(L')$, $q(L) \leq q(L')$, and $\bar{c}(L) \leq \bar{c}(L')$.*

Proof. Consider an arbitrary partial path ω starting at node $j(L) = j(L')$ and finishing at node i. If path $(\omega(L'), \omega)$ is feasible then path $(\omega(L), \omega)$ is also feasible due to conditions $j(L) = j(L')$, $J(L) \subseteq J(L')$, and $q(L) \leq q(L')$. The reduced cost of path $(\omega(L'), \omega)$ is not smaller than that of $(\omega(L), \omega)$, as the total reduced costs of arcs of the former is not smaller due to $\bar{c}(L) \leq \bar{c}(L')$, and the contribution of RLKCs to the reduced cost of $(\omega(L'), \omega)$ is not smaller than the one of $(\omega(L), \omega)$ due to $q(L) \leq q(L')$. Thus, L dominates L'. □

Theorem 6 essentially says that RLKCs do not change the dominance rule in the labeling algorithm. However, RLKCs do affect that algorithm in more subtle ways. The forward-backward route symmetry is exploited in the bi-directional labeling algorithm as follows. Every label L is extended only if $q(L) \leq Q/2$. After the label extension phase, the concatenation phase is performed, in which partial paths corresponding to two generated labels are concatenated to form a complete path. To speed up this concatenation phase, *completion bounds* are used. Given a node $j \in J$ and a set of labels $\mathcal{L}$ such that $j(L) = j$ for all $L \in \mathcal{L}$, completion bound $B_1(j, \mathcal{L})$ gives the minimum reduced cost of labels in $\mathcal{L}$: $B_1(j, \mathcal{L}) = \min_{L \in \mathcal{L}}\{\bar{c}(L)\}$. Due to the fact that function $g(q)$ is non-decreasing, value

$$\bar{c}(L') + g(q(L')) + B_1(j, \mathcal{L}) \tag{49}$$

gives a valid lower bound for the reduced cost of any complete path obtained by concatenation of paths $\omega(L')$ and $\omega(L)$ with $j(L') = j$ and $L \in \mathcal{L}$. If value (49) is not smaller than the minimum

reduced cost of a complete feasible path found so far, then concatenation of label L' with all labels in a set $\mathcal{L}$ may be skipped, saving significant time.

However, completion bounds $B(j, \mathcal{L})$ may not be tight as the reduced cost of labels do not include the contribution of RLKCs. We can reinforce completion bounds B_1 by defining $B_2(j, \mathcal{L}) = \min_{L \in \mathcal{L}}\{\bar{c}(L) + g(q(L))\}$. The total load of any concatenated path $(w(L'), w(L))$, where $j(L') = j(L)$, is equal to $q(L) + q(L')$. Thus $g(q(L) + q(L'))$, i.e., the contribution of RLKCs to the reduced cost of this path, is not smaller than $g(q(L)) + g(q(L')$ due to superadditivity of function $g(q)$. Therefore, value (49) in which B_1 is replaced by B_2 is still a valid lower bound for the reduced cost of any complete path obtained by concatenation of paths $\omega(L')$ and $\omega(L)$ with $j(L') = j$ and $L \in \mathcal{L}$.

Completion bounds are used not only in the concatenation phase of the labeling algorithm but also during the bucket arc elimination procedure, as well as in the elementary route enumeration procedure. So, the superadditivity property of RLKCs plays an important role in keeping those algorithms efficient.

6 Computational experiments

The algorithm is implemented in C++ language. We use the following third-party libraries and codes:

- BaPCod library (Sadykov and Vanderbeck, 2021) which implements the BCP framework;
- RCSP library, developed by Sadykov et al. (2021) which implements the bucket graph based labeling algorithm, bucket arc elimination procedure, elementary route enumeration, and also the separation algorithms for R1Cs and RCCs;
- IBM CPLEX Optimizer version 12.10 as the LP and MIP solver.

Experiments are run on a single thread of a 2x18-core Cascade Lake Intel Xeon Skylake Gold 6240 server at 2.6 GHz and having 196 GB of RAM.

6.1 Capacitated Location-Routing Instances

The proposed BCP algorithm is tested on instances from the literature. The first set of instances, which we denote as `PPW06`, was introduced by Prins et al. (2006) and contains instances with 20, 50, 100, or 200 customers, and with 5 or 10 possible depot locations. As we do not consider the four instances with only 20 customers, the set has 26 instances. The second set of instances, which we denote as `TB99`, was introduced by Tuzun and Burke (1999). Depots are uncapacitated in those instances. We only consider the nine instances, with 100 or 150 customers and 10 possible depot locations, that were also considered in Contardo et al. (2014): P111112, P111212, P112112, P112212, P113112, P113212, P131212, P131112, P131212, and P132112. The third set of instances, which we denote as `SL19`, was introduced by Schneider and Löffler (2019). We consider the instances with 100, 200 or 300 customers, and with 5, 10, 15, or 20 possible depot locations.

In the first experiment, we assess the impact of different families of cuts on the efficiency of our algorithm. That evaluation is performed on classic `PPW06` instances. We test the following variants of the BCP algorithm :

- BCP_0 — the base variant obtained by adapting in a straightforward way (by adding depot opening variables and depot capacity constraints) the best MDVRP algorithm in the literature (Sadykov et al., 2021) for the CLRP. Only two families of cuts are separated: rounded capacity cuts (RCC) and limited-memory rank-1 cuts (lm-R1C). Those cuts are generic for many VRP variants and do not take the particular structure of CLRP into account.

- BCP_{all} — the variant, in which all families of cuts described in this paper are used. This means that problem-specific RLKCs, DCCs, FCs, COVs, and GUBs are also separated.
- $\text{BCP}_{\text{all - CUT}}$ — the variant, in which all families of cuts are used except one (either RLKC, DCC, FC, COV, or GUB).

To exclude as much as possible the randomness related to finding good primal solutions earlier, all algorithm variants are given as initial primal bounds the values of the best-known solutions (the best solutions found in the recent literature, but also the improving solutions obtained during this work). The time limit is set to 12 hours.

Table 1 gives an overview of the performance of seven variants of our algorithm. The columns give the variant, average primal-dual relative gap after solving the root node, the geometric mean of the time needed for solving the root node (in seconds), the average number of branch-and-bound nodes, the geometric mean and the arithmetic mean (average) of total solution time in seconds, and the number of instances solved within the time limit. The total solution time is equal to the time limit for the unsolved instances.

In Figure 1, we also give the performance profile for all the tested variants of the BCP algorithm. Each point (X, Y) of the line corresponding to a variant says that for Y instances the solution time for this variant is not more than X times larger than the minimum solution time for all variants. So, the higher is the line corresponding to a variant, the more efficient is this variant. The horizontal axis in that graphic is logarithmic.

Table 1: Comparison of variants of the BCP algorithm on CLRP instances `PPW06`

	Root			Average	Geomean	
Variant	Gap	Time (s)	Nodes	Time (s)	Time (s)	Solved
BCP_{0}	4.46%	57.9	19.2	6373.8	758.7	24/26
$\text{BCP}_{\text{all}-\text{GUB}}$	3.08%	99.0	9.0	3314.6	481.0	25/26
$\text{BCP}_{\text{all}-\text{DCC}}$	0.85%	101.0	9.1	5995.1	504.6	24/26
$\text{BCP}_{\text{all}-\text{FC}}$	0.67%	111.4	4.4	3174.7	283.9	25/26
$\text{BCP}_{\text{all}-\text{RLKC}}$	0.52%	114.7	4.1	2994.8	264.0	25/26
$\text{BCP}_{\text{all}-\text{COV}}$	0.49%	114.4	4.6	3172.6	273.4	25/26
BCP_{all}	0.48%	115.0	4.1	3109.0	265.5	25/26

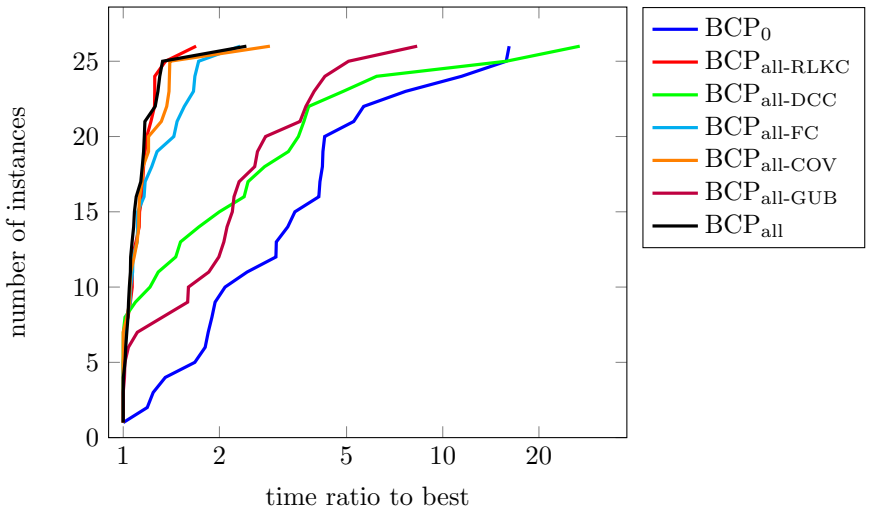

Figure 1: Performance profiles of BCP variants on CLRP instances `PPW06`

From the results, we can see that adding all problem-specific cuts makes the BCP algorithm

significantly more efficient than the base variant. Among individual families of cuts, DCCs have the most impact on the mean solution time, and allow one to solve one more instance. GUBs have the most impact on the average root gap. The impact of the family FC is smaller. The families COV and RLKC have a modest impact both on the root gap and on the total solution time.

We believe that the efficiency of RLKCs depends on the instance characteristics, and in particular on the ratio ρ between the vehicle capacity and the average depot capacity: $\rho = \frac{Q}{\sum_{i \in I} W_i / |I|}$. Our hypothesis is that the larger is value ρ, the larger should be the impact of the family RLKC. In fact, ratio ρ is rather small for instances `PPW06`, and it decreases with the increase of the instance size. It is on average 0.3 for instances with 50 customers, 0.18 for instances with 100 customers, and 0.1 for instances with 200 customers. Thus, a small impact of RLKCs is expected.

To test this hypothesis, we generated additional instances which are based on those in the set `PPW06`, but with ratio ρ equal to 0.3, 0.5, and 0.7. The procedure to modify an original instance with value ρ to obtain the instance with desired ratio ρ' is the following. First, we multiply the capacity and the cost of depots by factor σ: $f'_i = \lceil \sigma f_i \rceil$, and $W'_i = \lceil \sigma W_i \rceil$ for all $i \in I$, where

$$\sigma = \max \left\{ \frac{\rho}{\rho'}, \frac{\sum_{j \in J} d_j}{\sum_{i \in I} W_i - \max_{i \in I} W_i} \right\}.$$

Value σ is calculated in such a way that, for any depot $i \in I$, there exists a feasible solution to the modified instance with depot i closed, i.e. opening of any depot cannot be fixed by preprocessing. In the case $\sigma > \rho/\rho'$, we additionally increase the vehicle capacity: $Q' = \lceil Q\rho\sigma/\rho' \rceil$.

The modified instances are more difficult than the original ones: we could not obtain good feasible solutions for instances with 200 customers. Thus, we consider only modified instances with 50 and 100 customers. In total, we use 20 instances with $\rho = 0.3$, 17 instances with $\rho = 0.5$, and 17 instances with $\rho = 0.7$. The different number of instances is explained by the fact that sometimes the same instance is obtained by modification procedure from two different original instances. Such duplicated instances are removed.

We test four variants of our algorithm: BCP_0, BCP_{all}, $\text{BCP}_{\text{all - RLKC}}$, and $\text{BCP}_{\text{all - DCC}}$ to access the impact of cut families DCC and RLKC on instances with different values ρ. We use the value of the best solution we were able to obtain during the preliminary experiments as the initial primal bound. For 51 instances out of 54, we were able to obtain the optimal solution value. The time limit is set to 12 hours. Table 2 gives the performance of the variants of our algorithm separately on modified instances `PPW06` for different value ρ. The meaning of the columns is the same as in Table 1.

Table 2: Comparison of variants of the BCP algorithm on modified instances `PPW06` with different value ρ.

		Root			Average	Geomean	
Variant	ρ	Gap	Time (s)	Nodes	Time (s)	Time (s)	Solved
BCP_0	0.3	2.14%	32.1	18.6	4290.3	344.7	19/20
$\text{BCP}_{\text{all}-\text{DCC}}$	0.3	0.74%	41.0	12.7	4203.0	260.9	19/20
$\text{BCP}_{\text{all}-\text{RLKC}}$	0.3	0.51%	48.2	6.1	3148.5	174.1	19/20
BCP_{all}	0.3	0.46%	49.3	5.8	3057.3	162.9	19/20
BCP_0	0.5	3.33%	42.9	76.7	15873.3	2513.6	13/17
$\text{BCP}_{\text{all}-\text{DCC}}$	0.5	2.09%	108.4	35.0	12963.9	1979.7	13/17
$\text{BCP}_{\text{all}-\text{RLKC}}$	0.5	1.73%	69.8	24.3	11155.1	1059.3	14/17
BCP_{all}	0.5	1.26%	120.3	13.2	8248.7	813.6	15/17
BCP_0	0.7	5.94%	51.6	255.3	28272.8	10511.0	6/17
$\text{BCP}_{\text{all}-\text{DCC}}$	0.7	2.49%	247.7	58.1	16503.2	4531.6	12/17
$\text{BCP}_{\text{all}-\text{RLKC}}$	0.7	3.91%	83.0	89.4	22259.4	5438.7	10/17
BCP_{all}	0.7	1.53%	284.6	18.9	9491.4	1734.7	14/17

The results presented in Table 2 confirm our hypothesis. Impact on the BCP performance of the family RLKC remains marginal for instances with $\rho = 0.3$, whereas it becomes noticeable for instances with $\rho = 0.5$. For instances with $\rho = 0.7$, this impact is instrumental, as the employment of RLKCs allows us to decrease the root gap by more than half, divide the number of nodes by almost five, divide the solution time by three, and solve four more instances to optimality.

In Table 3, we show the average number of generated cuts of each family (and the average number of active cuts at the end of the root node in brackets). These statistics are shown both for the original instances `PPW06`, as well as for the modified instances.

Table 3: Cut generation statistics for original and modified instances `PPW06`.

	Original	Modified instances		
Cut family	instances	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$
RCC	492.5 (11.6)	349.5 (7.2)	366.8 (2.9)	254.9 (1.9)
lm-R1C	7044.0 (215.4)	22405.5 (153.0)	35692.0 (223.8)	35610.9 (236.2)
COV	30.8 (0.2)	28.1 (0.1)	40.1 (0.3)	30.7 (0.1)
FC	4.0 (0.7)	3.9 (0.6)	8.5 (0.6)	4.4 (0.5)
GUB	338.5 (78.4)	282.4 (30.0)	228.8 (20.9)	203.2 (12.9)
DCC (total)	488.4 (9.9)	1135.1 (10.1)	1341.9 (14.8)	1636.1 (11.4)
DCC1	419.1 (8.9)	1078.0 (9.2)	1320.3 (14.8)	1622.4 (11.4)
DCC2	69.3 (1.0)	57.1 (0.8)	21.6 (0.0)	13.7 (0.0)
RLKC (total)	53.4 (1.2)	325.1 (3.9)	11726.6 (17.9)	12785.2 (29.6)
RLKCround	26.8 (0.7)	296.1 (3.7)	1304.8 (4.1)	109.9 (0.8)
RLKC1/2	0.7 (0.0)	0.5 (0.0)	757.2 (1.8)	1401.8 (7.7)
RLKC1/3	0.7 (0.0)	0.8 (0.0)	395.6 (0.4)	675.7 (1.6)
RLKC1/4	1.2 (0.0)	1.4 (0.1)	362.7 (0.8)	418.2 (0.9)
RLKC1/5	1.8 (0.0)	1.5 (0.0)	1115.7 (0.9)	1738.4 (3.8)
RLKC1/6	2.6 (0.0)	3.2 (0.1)	780.2 (1.9)	843.4 (1.7)
RLKC1/8	5.7 (0.1)	5.0 (0.1)	2318.9 (3.3)	2611.1 (4.9)
RLKC1/10	13.8 (0.3)	16.6 (0.1)	4691.6 (4.9)	4986.7 (8.2)

In the next experiment, we compare our algorithm BCP_{best} with the algorithm by Contardo et al. (2014) on instances `TB99` and `PPW06`. For a fair comparison, we use here as initial primal bounds the same values employed in Contardo et al. (2014). We set a time limit of 30 hours for our algorithm. An aggregated comparison is presented in Table 4. For each algorithm, we give the arithmetic and geometric mean times in seconds and the number of instances solved to optimality. The solutions times of Contardo et al. (2014) are normalized using the CPU marks provided by PassMark Single Thread Performance (`https://www.cpubenchmark.net/singleThread.html`), so they are comparable to our times.

Table 4: Comparison of BCP_{best} with Contardo et al. (2014) on instances in the sets `TB99` and `PPW06`

	BCP_{all}			Contardo et al. (2014)		
Instance set	Solved	Av. Time	Gm. Time	Solved	Av. Time	Gm. Time
`PPW06`	24/26	14768	518	16/26	14235	836
`TB99`	9/9	4834	945	6/9	46290	5589

It can be seen from Table 4 that the new algorithm can solve 11 instances that could not be solved in Contardo et al. (2014). We remark that one additional instance `coord200-10-2` can be solved by the new algorithm in less than six hours, if improved initial upper bounds (that were not available to Contardo et al. (2014)) are used. The only open instance in set `PPW06` is now `coord200-10-3b`. We also evaluate the performance of our BCP algorithm when no initial

primal bound is passed to it. In this case, still, 24 out of 26 instances are solved to optimality within the time limit of 30 hours. The geometric mean of the solution time is increased from 518 seconds to 935 seconds, and the geometric mean of the number of nodes is increased from 7.7 to 11.7. All detailed results for instances `PPW06` and `TB99` are reported in E-Companion B.

Finally, we evaluate the performance of our algorithm on recently introduced instances `SL19`. This is the first exact algorithm applied for these instances. We use here as initial primal bounds the best-known solutions found by Schneider and Löffler (2019). The time limit is set to 30 hours. The instances are divided into groups depending on their size, i.e., the number of potential depot locations $|I|$ and the number of customers $|J|$. The results for each group of instances are shown in Table 5. The columns give the size of instances, the number of instances solved to optimality, the number of instances for which we could find improving solutions, and the average improvement for these instances.

Table 5: Performance of BCP_{best} on instances in the set `SL19`

Instances				
$\|I\|$	$\|J\|$	Solved	Improved BKS	Improvement
5	100	14/14	7/14	0.05%
10	100	14/14	5/14	0.11%
10	200	11/14	13/14	0.08%
15	200	15/20	18/20	0.12%
15	300	6/20	11/20	0.29%
20	300	4/20	8/20	0.91%

As it can be seen from Table 5, our algorithm could solve to optimality all instances with 100 customers, three quarters of instances with 200 customers, but only one quarter of the instances with 300 customers. We could improve the best-known solutions for 62 instances. The average improvement is small for instances with up to 15 depots, but it becomes very significant for instances with 300 customers and 20 depots. For the instance `300-20-1e`, the improvement exceeds 5%. The results indicate that the heuristic of Schneider and Löffler (2019) obtains solutions of excellent quality for instances with up to 15 depots. That quality decreases for instances with 20 depots. The root gap (from the best-known solution) of our BCP algorithm for the largest instances sometimes reaches 6–8%. Whereas for instances solved to optimality a typical root gap is below 1% and never exceeds 2%. This suggests that some best-known solutions for instances with 300 customers and 20 depots may be far away from the optimal ones. Detailed results for instances `SL19` are reported in E-Companion B.

6.2 VRP-CMD instances

In this section, we test our algorithm on VRP-CMD instances which arise when solving the cut generation subproblem of the two-echelon stochastic multi-period capacitated location-routing problem by a logic-based Benders decomposition approach (Ben Mohamed et al., 2020). We have randomly selected 199 instances, which have 50 customers and from three to five depots.

Despite the fact the VRP-CMD is very similar to the standard multi-depot vehicle routing problem (MDVRP), the introduction of tight depot capacities makes the problem much more difficult. MDVRP instances from the literature with less than 80 customers are consistently solved to optimality in seconds, often in the root node (Sadykov et al., 2021). On the other hand, the VRP-CMD instances we consider here can take many minutes (or even hours) and require the exploration of big branch-and-bound trees. Root gaps are very large and may reach 15% of the optimal value.

In this experiment, we computationally estimate the impact of valid inequalities on the efficiency of the BCP algorithm. As in the VRP-CMD all depots are considered open, variables y are fixed to one and valid inequalities FC, COV, and GUB are not useful. Thus, we test the

following BCP variants.

- BCP_0 — the base variant obtained by adapting in a straightforward way, by only adding the depot capacity constraints, the best MDVRP algorithm in the literature (Sadykov et al., 2021) for the VRP-CMD. Again, only RCCs and lm-R1Cs are separated.
- BCP_{0+RLCK} — the base variant with additional separation of RLKCs.
- BCP_{0+DCC} — the base variant with additional separation of DCCs.
- BCP_{all} — the variant with separation of all valid inequalities (RCC, lm-R1C, DCC, and RLKC).

To exclude the randomness related to finding good primal solutions earlier or later during the branch-and-bound, for this experiment we use only the 183 instances which we were able to solve to optimality during preliminary tests. The initial upper bound is set to that optimal solution value augmented by a small epsilon. The time limit is set to 3 hours. Table 6 presents the results for each of the four BCP variants tested. The columns give the average relative root gap from the optimal solution value, the geometric mean value for the root solution time, the average number of branch-and-bound nodes, the geometric mean value for the total solution time, and the number of instances solved. Detailed results on those 183 instances for variant BCP_{all} are reported in C.

Table 6: Comparison of variants of the BCP algorithm on VRP-CMD instances

	Root			Average	Geomean	
Variant	gap	time	Nodes	Time (s)	Time (s)	Solved
BCP_0	6.56%	12.0	194.0	3422.4	1502.8	156/183
BCP_{0+RLKC}	6.41%	14.8	59.3	1970.3	922.8	177/183
BCP_{0+DCC}	3.67%	22.2	58.0	1847.9	672.4	177/183
BCP_{all}	3.57%	26.5	25.7	1175.3	512.0	183/183

In addition to Table 6, we also give the performance profiles for the four tested BCP variants in Figure 2. Each line corresponds to one variant of the algorithm and depicts the number of instances solved within a given time expressed in minutes.

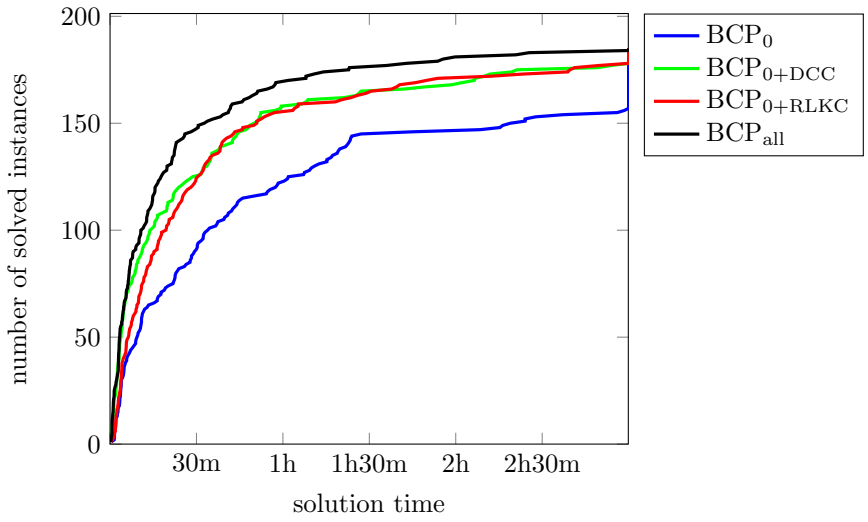

Figure 2: Performance profile for BCP variants tested on 183 VRP-CMD instances with known optimal solutions

Table 6 and Figure 2 show that both families DCC and RLKC have a very significant positive impact on the efficiency of the BCP algorithm. DCCs decrease the most the root gap, the number of nodes, and the average solution time. However, RLKCs have a larger impact on the number of solved instances. Clearly, the best variant of the BCP algorithm is the one that uses both families of cuts.

E-Companion C also reports results on the remaining 16 instances for BCP_{all}, which are run with the best-known solution values as initial upper bounds. Nine of these instances could be solved to optimality within 30 hours, and seven instances remain open.

6.3 VRPTW with Shifts instances

In this section, we test our BCP algorithm on instances of the VRPTW with Shifts. These instances were introduced by Dabia et al. (2019), who built them on top of the well-known Solomon instances for the VRPTW. There are instances with three different sizes: 25, 50, and 100 customers. All instances have three shifts. For each original Solomon instance, three instances were generated with three different shift capacities. For each size and each shift capacity, there are 56 instances, divided into classes `c1`, `c2`, `r1`, `r2`, `rc1`, `rc2`. Thus, in total there are 504 instances.

Shifts in this problem are modeled as capacitated depots. Again, as in the case of the VRP-CMD, there is no fixed cost to "open" a shift. Therefore, variables y are fixed to one and valid inequalities FC, COV, and GUB inequalities are not useful.

We run the variant BCP_{all} without initial upper bounds with the time limit of 30 hours. 474 out of 504 instances were solved to optimality, including all instances with 25 customers, and all but one instance with 50 customers. The detailed results are given in E-Companion D.

To compare the efficiency of our algorithm with the one proposed by Dabia et al. (2019), we need to calculate the number of solved instances within 30 minutes to take into account the difference in the computers used. Our algorithm solved 421 instances to optimality in 30 minutes including all instances with 25 customers. Considering that the algorithm by Dabia et al. (2019) solved 280 instances to optimality, we can say that our approach is considerably better. The best-known solutions are improved for 238 out of 504 instances.

Now, as in the previous section, we test four BCP variants: BCP_0, $\text{BCP}_{0+\text{RLCK}}$, $\text{BCP}_{0+\text{DCC}}$, and BCP_{all}. Again, to exclude the randomness related to finding good primal solutions earlier or later during the branch-and-bound, for this experiment we use only the 474 instances for which we know the optimal solutions. The time limit is set to three hours.

Table 7 presents the results separately for every instance size. The columns give the average relative root gap, geometric mean root solution time in seconds, the average number of branch-and-bound nodes, the geometric mean total solution time in seconds, and the number of instances solved.

In Figure 3, we also give the performance profiles for the four tested BCP variants applied to instances with 100 customers. Each line corresponds to one variant of the algorithm and depicts the number of instances solved within a given time expressed in minutes.

The results show a very significant positive impact of RLKCs, as their separation improves the solution time, the number of branch-and-bound nodes, and the number of solved instances. The separation of DCCs has also more modest, but still positive impact. This can be clearly seen in the performance profile in Figure 3.

Table 7: Comparison of variants of the BCP algorithm on VRPTW with Shifts instances

		Root			Average	Geomean	
Variant	$\|J\|$	Gap	Time (s)	Nodes	Time (s)	Time (s)	Solved
BCP_0	25	1.40%	1.1	1.2	3.5	1.2	168/168
BCP_{0+DCC}	25	0.25%	1.1	1.0	2.2	1.1	168/168
BCP_{0+RLKC}	25	0.02%	1.3	1.0	3.3	1.2	168/168
BCP_{all}	25	0.00%	1.1	1.0	2.0	1.0	168/168
BCP_0	50	1.96%	17.7	3.1	588.3	36.2	163/167
BCP_{0+DCC}	50	1.21%	15.5	2.0	354.0	23.8	166/167
BCP_{0+RLKC}	50	0.75%	24.7	1.4	384.6	28.8	164/167
BCP_{all}	50	0.50%	18.0	1.2	192.6	19.4	167/167
BCP_0	100	1.15%	90.7	7.0	2158.5	304.8	123/139
BCP_{0+DCC}	100	0.84%	112.2	4.3	1631.9	268.1	130/139
BCP_{0+RLKC}	100	0.56%	114.8	2.5	953.1	175.1	136/139
BCP_{all}	100	0.51%	124.9	2.3	885.2	175.4	137/139

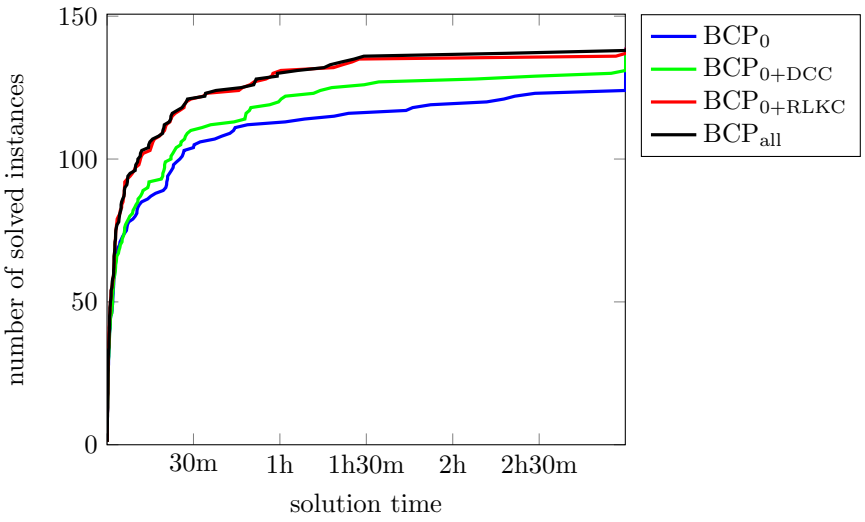

Figure 3: Performance profile for BCP variants tested on 100 customer instances with known optimal solutions of the VRPTW with Shifts

7 Conclusion

In this work we propose a BCP algorithm for the CLRP and for two other related problems with the nested knapsack structure, VRP-CMD and VRPTW-S.

The proposed algorithm for CLRP is clearly superior to the other exact algorithms in the literature. An important observation is that this is the first BCP algorithm that handles CLRP directly, instead of reducing it to the solution of a set of VRP-CMD subproblems obtained by fixing the opened depots. We believe that this direct approach has more potential for solving instances with many depot locations. In fact, instances with 15 and 20 depot locations could be solved. Results on VRPTW-S are also clearly better than those from the exact algorithms found in the literature.

The most original methodological contribution of this work is the introduction of RLKCs, a family of non-robust cuts derived from the "outer" knapsack constraints, the ones that are defined directly over the route variables. Those cuts are strong in the sense that they contain all the facets of the master knapsack polytope, dominating the cover cuts proposed in Dabia et al. (2019). Some theoretical properties of those facets, monotonicity and superadditivity, are explored when adapting the labeling algorithm (used in the pricing) for handling RLKCs,

keeping it efficient. In fact, the adaptation was so successful that the BCP does not need any mechanism for limiting or controlling the RLKCs, they are treated like robust cuts. The overall positive impact of RLKCs in the BCP performance varied, depending on the tested problem and even on the characteristics of the instances. For the CLRP, the RLKCs proved to be effective on instances with tight depot capacities. As those instances are harder, the RLKCs make the final BCP algorithm significantly more robust.

In future works, the BCP algorithm proposed in this work could be applied to other problems with the nested knapsack structure, like the capacity and distance-constrained plant location problem (Albareda-Sambola et al., 2009) or the last-mile vehicle routing problem with delivery options (Tilk et al., 2021), mentioned in the literature review. Other variants of the LRP, like the LRP with time windows (Ponboon et al., 2016) or the two-echelon LRP (Contardo et al., 2012) could also be approached.

Acknowledgements

Author EU was partially supported by the following grants: CNPq 313601/2018-6, Faperj E-26/202.887/2017, and CAPES PrInt UFF no 88881.

A Proof of Theorem 3

Proof. If $\xi t \leq 1$ defines facet of $\mathcal{P}_{\text{MKP}}(W_i)$ there are W_i affinely independent points p^j, $j = 1, \ldots, |W_i|$, satisfying $\xi p^j = 1$. Then $(p^j, 1)$, $j = 1, \ldots, |W_i|$, and $(\mathbf{0}, 0)$ are $W_i + 1$ affinely independent points satisfying $\xi\theta = y_i$.

If $\xi\theta \leq y_i$ defines a facet of $\text{conv}\{(\theta^i, y_i) \in \mathbb{Z}_+^{W_i} \times \{0,1\} : \sum_{q=1}^{W_i} q\,\theta_q \leq W_i y_i\}$ there are $W_i + 1$ affinely independent integer points satisfying $\xi\theta = y_i$. One of those points is certainly $(\mathbf{0}, 0)$, the remaining points have the format $(p^j, 1)$, $j = 1, \ldots, |W_i|$. Then, p^j, $j = 1, \ldots, |W_i|$, are W_i affinely independent points satisfying $\xi\theta = 1$. □

B Detailed results for LRP

In the following tables, BKS stands for the best-known solution, IPB for the initial primal bound, RDB for the root dual bound, Rg for the root gap, Rt for the time spent in the root node, Nodes for the number of nodes explored in the branch-and-bound tree, Fg for the final gap, BPB for the best primal bound, and t for the total time. New best solutions are in bold.

In Table 8, the IPB values that were already proven to be optimal are labeled with *. Nine instances were solved to optimality for the first time. In two cases the optimal solutions improve upon the previous best-known solutions. Table 9 are similar experiments, but without using IPBs. The results show that one less instance is solved. For the 24 instances solved in both ways, the geometric mean of the running times increases from 238 seconds to 628 seconds, a factor of 2.6. It is clear that external bounds do help the BCP algorithm, but they are not essential for overall performance. Table 10 and Table 11 are comparisons with Contardo et al. (2014), using the same IPBs. Finally, Table 12 are results on the new Schneider and Löffler (2019) instances, using the best-known solutions in the literature as IPBs.

Table 8: Detailed results for algorithm BCP_{all} initialized with the best known primal bounds in the literature on instances `PPW06`

Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)
coord50-5-1	90111*	0.00%	8	1	0.00%	90111	7
coord50-5-1b	63242	0.00%	84	1	0.00%	63242	83

Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)
coord50-5-2	88298*	2.22%	10	3	0.00%	88298	14
coord50-5-2BIS	84055*	0.00%	5	1	0.00%	84055	5
coord50-5-2b	67308*	2.37%	74	3	0.00%	67308	174
coord50-5-2bBIS	51822*	0.00%	8	1	0.00%	51822	7
coord50-5-3	86203*	0.00%	19	1	0.00%	86203	18
coord50-5-3b	61830*	0.00%	38	1	0.00%	61830	37
coord100-10-1	287661	1.04%	161	55	0.00%	287661	2683
coord100-10-1b	230989	0.43%	360	13	0.00%	230989	2025
coord100-10-2	243590*	1.07%	153	9	0.00%	243590	660
coord100-10-2b	203988*	0.35%	457	3	0.00%	203988	936
coord100-10-3	250882	2.24%	137	33	0.00%	250882	1591
coord100-10-3b	203114	1.56%	214	31	0.00%	203114	11173
coord100-5-1	274814*	0.00%	32	1	0.00%	274814	31
coord100-5-1b	213568*	0.16%	272	3	0.00%	213568	393
coord100-5-2	193671*	0.00%	19	1	0.00%	193671	18
coord100-5-2b	157095*	0.00%	157	1	0.00%	157095	156
coord100-5-3	200079*	0.00%	64	1	0.00%	200079	63
coord100-5-3b	152441*	0.00%	90	1	0.00%	152441	90
coord200-10-1	474850	0.25%	481	21	0.00%	**474702**	1949
coord200-10-1b	375177	0.09%	2842	5	0.00%	375177	4354
coord200-10-2	448077	0.27%	586	213	0.00%	**448005**	20961
coord200-10-2b	373696	0.12%	1907	9	0.00%	373696	6633
coord200-10-3	469433*	0.07%	472	3	0.00%	469433	578
coord200-10-3b	362320	0.39%	1755	135	0.21%	362320	107998

Table 9: Detailed results for algorithm BCP_{all} executed on instances `PPW06` without initial upper bound

Instance	BKS	RDB	Rt (s)	Nodes	Fg (%)	BPB	t (s)
coord50-5-1	90111	89853	15	3	0.00%	90111	38
coord50-5-1b	63242	62604	53	5	0.00%	63242	284
coord50-5-2	88298	86315	15	5	0.00%	88298	82
coord50-5-2BIS	84055	83875	17	3	0.00%	84055	74
coord50-5-2b	67308	65397	34	5	0.00%	67308	274
coord50-5-2bBIS	51822	51763	20	3	0.00%	51822	41
coord50-5-3	86203	85360	17	3	0.00%	86203	60
coord50-5-3b	61830	61329	38	3	0.00%	61830	99
coord100-10-1	287661	284453	93	79	0.00%	287661	4258
coord100-10-1b	230989	229802	177	15	0.00%	230989	3760
coord100-10-2	243590	240915	97	11	0.00%	243590	795
coord100-10-2b	203988	202990	198	7	0.00%	203988	1393
coord100-10-3	250882	245041	96	129	0.00%	250882	5630
coord100-10-3b	203114	199867	161	39	0.00%	203114	11517
coord100-5-1	274814	273995	29	3	0.00%	274814	79
coord100-5-1b	213568	212635	72	15	0.00%	213568	1620
coord100-5-2	193671	193506	37	3	0.00%	193671	63
coord100-5-2b	157095	156758	123	5	0.00%	157095	400
coord100-5-3	200079	199503	55	7	0.00%	200079	248
coord100-5-3b	152441	152344	191	3	0.00%	152441	355
coord200-10-1	474702	473451	290	57	0.00%	474702	7732
coord200-10-1b	375177	373958	956	31	0.00%	375177	19457
coord200-10-2	448077	446558	330	901	0.30%	448475	107995
coord200-10-2b	373696	372523	715	51	0.00%	373696	29804
coord200-10-3	469433	468846	389	9	0.00%	469433	2274
coord200-10-3b	362320	360656	996	43	–	–	108148

Table 10: Detailed comparison between algorithm BCP_{all} and the approach by Contardo et al. (2014) on instances PPW06

		BCP_{best}						Contardo et al. (2014)	
Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)	BPB	t (s)
coord50-5-1	90111	0.00%	8	1	0.00%	90111	7	90111*	9
coord50-5-1b	63242	0.00%	83	1	0.00%	63242	83	63242*	243
coord50-5-2	88298	2.22%	10	3	0.00%	88298	14	88298*	7
coord50-5-2BIS	84055	0.00%	5	1	0.00%	84055	5	84055*	15
coord50-5-2b	67340	2.42%	74	3	0.00%	67308	179	67308*	226
coord50-5-2bBIS	51822	0.00%	8	1	0.00%	51822	8	51822*	4
coord50-5-3	86203	0.00%	19	1	0.00%	86203	19	86203*	48
coord50-5-3b	61830	0.00%	37	1	0.00%	61830	37	61830*	3
coord100-10-1	289017	1.52%	140	117	0.00%	287661	5259	289017	23543
coord100-10-1b	234641	2.01%	259	15	0.00%	230989	4092	234641	19351
coord100-10-2	243590	1.07%	149	7	0.00%	243590	516	243590*	1144
coord100-10-2b	203988	0.35%	454	3	0.00%	203988	931	203988*	165
coord100-10-3	252421	2.85%	134	123	0.00%	250882	5255	252421	12656
coord100-10-3b	204597	2.28%	200	285	0.00%	203114	71156	204597	40740
coord100-5-1	274814	0.00%	31	1	0.00%	274814	31	274814*	227
coord100-5-1b	214392	0.65%	188	7	0.00%	213568	814	213568	24306
coord100-5-2	193671	0.00%	19	1	0.00%	193671	18	193671*	29
coord100-5-2b	157173	0.15%	163	3	0.00%	157095	229	157095*	6389
coord100-5-3	200079	0.00%	65	1	0.00%	200079	64	200079*	63
coord100-5-3b	152441	0.00%	89	1	0.00%	152441	89	152441*	239
coord200-10-1	479425	1.22%	364	101	0.00%	474702	13293	479425	25298
coord200-10-1b	378773	1.17%	1529	37	0.00%	375177	20533	378773	120447
coord200-10-2	450468	0.82%	423	1049	0.28%	448409	107997	450468	15871
coord200-10-2b	374435	0.35%	1369	81	0.00%	373696	41884	374435	46988
coord200-10-3	472898	0.84%	488	17	0.00%	469433	3341	469433*	9519
coord200-10-3b	364178	0.95%	1162	91	0.71%	364178	108107	364178	22867

Table 11: Detailed comparison between algorithm BCP_{best} and the approach by Contardo et al. (2014) on instances TB99

		BCP_{best}						Contardo et al. (2014)	
Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)	BPB	t (s)
coordP111112	1467.68	0.00%	136	1	0.00%	1467.68	134	1467.68*	2269
coordP111212	1394.80	0.36%	315	3	0.00%	1394.80	521	1394.80*	10947
coordP112112	1167.16	0.00%	257	1	0.00%	1167.16	256	1167.16*	158
coordP112212	791.66	0.00%	398	1	0.00%	791.66	397	791.66*	934
coordP113112	1245.45	1.92%	2442	19	0.00%	1238.24	28648	1238.24	87225
coordP113212	902.26	0.00%	130	1	0.00%	902.26	129	902.26*	78
coordP131112	1900.70	0.57%	1500	5	0.00%	1892.17	3628	1896.98	177697
coordP131212	1965.12	0.77%	698	19	0.00%	1960.02	9097	1965.12	128169
coordP132112	1443.33	0.00%	696	1	0.00%	1443.32	693	1443.32*	9451

Table 12: Detailed results for algorithm BCP_{all} initialized with best known primal bounds in the literature on instances SL19

Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)
100-5-1c	134516	0.00%	20	1	0.00%	134516	19
100-5-1d	275749	0.32%	48	5	0.00%	275749	121
100-5-1e	292311	0.00%	41	1	0.00%	**292301**	40
100-5-2c	83989	0.16%	9	1	0.00%	**83855**	9
100-5-2d	242266	0.88%	67	3	0.00%	**242105**	134
100-5-2e	253888	0.30%	84	9	0.00%	253888	253
100-5-3c	87555	0.00%	12	1	0.00%	87555	11

Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)
100-5-3d	226783	0.07%	56	1	0.00%	**226634**	55
100-5-3e	252603	0.00%	50	1	0.00%	252603	50
100-5-4a	255853	0.00%	30	1	0.00%	255853	29
100-5-4b	214425	0.00%	94	1	0.00%	214425	94
100-5-4c	98129	0.53%	59	5	0.00%	**98104**	124
100-5-4d	250315	0.16%	61	5	0.00%	**250301**	99
100-5-4e	211159	0.85%	116	479	0.00%	**211113**	15045
100-10-1c	92629	0.00%	17	1	0.00%	92629	16
100-10-1d	363930	0.50%	90	31	0.00%	363930	1176
100-10-1e	344322	0.54%	191	11	0.00%	344322	536
100-10-2c	84717	0.00%	17	1	0.00%	84717	16
100-10-2d	343252	1.14%	88	53	0.00%	343252	2404
100-10-2e	332900	0.20%	186	3	0.00%	332900	275
100-10-3c	85618	0.29%	41	1	0.00%	**85369**	40
100-10-3d	329990	1.28%	78	443	0.00%	329990	13177
100-10-3e	318156	0.88%	167	29	0.00%	**318109**	1508
100-10-4a	253892	0.69%	262	9	0.00%	**253471**	607
100-10-4b	211354	0.65%	743	5	0.00%	211354	2154
100-10-4c	86215	0.00%	31	1	0.00%	86215	30
100-10-4d	328251	1.35%	117	2267	0.00%	**328181**	74215
100-10-4e	308866	0.55%	196	455	0.00%	**308757**	19460
200-10-1c	156087	0.34%	325	5	0.00%	**156029**	564
200-10-1d	638452	0.40%	398	111	0.00%	**638068**	10811
200-10-1e	599463	0.86%	694	141	0.00%	**599069**	14871
200-10-2c	144337	0.51%	379	83	0.00%	**144046**	8427
200-10-2d	663814	0.46%	586	823	0.23%	**663154**	107995
200-10-2e	619037	0.25%	1381	467	0.00%	**618858**	53043
200-10-3c	184885	0.06%	458	1	0.00%	**184783**	452
200-10-3d	640357	0.06%	451	3	0.00%	**640289**	587
200-10-3e	604617	0.13%	1019	95	0.00%	**604480**	9448
200-10-4a	452870	0.99%	868	835	0.00%	**452430**	85744
200-10-4b	369951	0.78%	2284	135	0.43%	**369871**	107997
200-10-4c	144407	0.46%	400	125	0.00%	**144013**	10930
200-10-4d	618590	0.34%	499	297	0.00%	**617932**	37803
200-10-4e	562854	0.36%	1114	893	0.25%	562854	107995
200-15-1a	461203	0.38%	748	153	0.00%	**460430**	18752
200-15-1b	367397	0.54%	1872	113	0.00%	**366359**	67910
200-15-1c	148218	0.05%	134	1	0.00%	**148141**	127
200-15-1d	813941	0.24%	424	25	0.00%	**813576**	2225
200-15-1e	708837	2.15%	686	1181	0.26%	**708585**	107993
200-15-2a	513893	1.97%	608	211	0.00%	**513512**	21920
200-15-2b	406843	0.50%	2187	137	0.18%	**406839**	107992
200-15-2c	135051	0.30%	316	3	0.00%	**134779**	336
200-15-2d	811722	0.29%	776	41	0.00%	**811361**	4805
200-15-2e	712524	0.56%	877	649	0.34%	712524	107993
200-15-3a	455676	0.53%	1249	151	0.00%	**455351**	20594
200-15-3b	357086	0.55%	2668	65	0.00%	**356887**	71800
200-15-3c	141129	0.56%	475	11	0.00%	**140765**	1216
200-15-3d	877638	0.31%	574	111	0.00%	**877543**	13590
200-15-3e	816377	2.73%	828	809	2.23%	**816129**	107994
200-15-4a	433268	0.44%	668	513	0.00%	**432672**	51175
200-15-4b	349269	0.91%	1896	137	0.50%	349269	107994
200-15-4c	143772	0.81%	570	27	0.00%	**143052**	3196
200-15-4d	828144	0.22%	473	179	0.00%	**826829**	33505
200-15-4e	700202	1.53%	1649	79	0.00%	**700013**	13447
300-15-1a	856267	0.66%	3135	389	0.25%	**854503**	107989
300-15-1b	622412	0.91%	9755	5	0.62%	622412	108004
300-15-1c	366770	0.70%	1124	35	0.00%	**364979**	6341
300-15-1d	1339010	0.47%	1582	409	0.33%	**1338255**	107987
300-15-1e	1217690	1.34%	2645	275	0.63%	1217690	107989
300-15-2a	759999	1.02%	3283	331	0.21%	**757931**	107990
300-15-2b	557912	0.60%	16604	13	0.44%	557912	107996
300-15-2c	311558	0.79%	1001	437	0.00%	**310061**	69751
300-15-2d	1301863	0.38%	1748	469	0.15%	**1301210**	107984
300-15-2e	1272700	0.70%	3191	371	0.62%	1272700	107983
300-15-3a	778023	0.36%	3311	39	0.00%	**776531**	20139
300-15-3b	594073	0.32%	23608	3	0.30%	594073	115651
300-15-3c	341712	0.64%	1931	23	0.00%	**340155**	5450
300-15-3d	1358223	0.83%	2162	261	0.66%	1358223	108036

Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)
300-15-3e	1286877	0.55%	4233	289	0.46%	1286877	108007
300-15-4a	747730	0.24%	3488	109	0.00%	**746407**	22202
300-15-4b	559877	0.54%	6247	45	0.47%	559877	107982
300-15-4c	304254	0.89%	1842	55	0.00%	**302390**	15149
300-15-4d	1288091	0.44%	2045	305	0.19%	**1285714**	107990
300-15-4e	1173516	1.92%	2362	221	0.58%	1173516	107989
300-20-1a	1009840	6.86%	3018	107	6.76%	1009840	108004
300-20-1b	739604	0.81%	12842	45	0.63%	739604	108023
300-20-1c	364096	0.92%	1502	21	0.00%	**361735**	6084
300-20-1d	1575390	0.50%	2601	353	0.23%	**1572968**	107978
300-20-1e	1391567	7.97%	1769	137	1.61%	**1320811**	107990
300-20-2a	909306	1.14%	4364	287	0.65%	909306	107979
300-20-2b	695524	0.86%	9876	23	0.46%	695524	108011
300-20-2c	299425	0.58%	1735	33	0.00%	**298522**	8333
300-20-2d	1569139	0.54%	2623	257	0.45%	**1569042**	107982
300-20-2e	1386386	7.88%	2043	107	7.75%	1386386	108070
300-20-3a	929901	0.80%	3786	149	0.00%	**927452**	62525
300-20-3b	751307	0.75%	11312	3	0.75%	751307	108014
300-20-3c	305771	0.88%	1846	71	0.00%	**304269**	23007
300-20-3d	1539008	0.65%	3376	215	0.54%	1539008	107977
300-20-3e	1289734	3.46%	3968	93	3.20%	1289734	107992
300-20-4a	859474	2.24%	3061	135	1.45%	859474	107979
300-20-4b	687930	1.09%	7554	25	0.91%	687930	108003
300-20-4c	300285	1.04%	1606	391	0.59%	**299206**	107981
300-20-4d	1540194	0.59%	2868	115	0.47%	1540194	108021
300-20-4e	1344056	3.17%	2578	103	2.72%	1344056	107983

C Detailed results for VRP-CMD

In the following tables, IPB stands for the initial primal bound, Rg for the root gap, Rt for the time spent in the root node, Nodes for the number of nodes explored in the branch-and-bound tree, Fg for the final gap, BPB for the best primal bound, and t for the total time.

Table 13 contains instances for which the initial primal bound is the optimal solution.

Table 13: Detailed results for algorithm BCP_{all} for 183 instances of the VRP-CMD with known optimal solutions

Instance	Rg (%)	Rt (s)	Nodes	OPT	t (s)
MDCVRP-10	4.91%	21	67	12173.31	556
MDCVRP-34	0.40%	40	237	8926.51	3434
MDCVRP-60	9.87%	20	105	10372.76	763
MDCVRP-62	4.50%	27	5	9106.99	44
MDCVRP-74	5.60%	13	17	11038.03	291
MDCVRP-96	5.88%	15	39	10986.37	1293
MDCVRP-97	8.02%	17	99	11581.48	1521
MDCVRP-99	9.61%	17	25	12139.98	413
MDCVRP-103	0.17%	31	21	9121.74	1276
MDCVRP-126	3.32%	27	61	11113.19	2545
MDCVRP-130	3.17%	35	9	10426.02	137
MDCVRP-157	5.74%	15	23	11070.27	870
MDCVRP-169	5.35%	23	241	11562.64	3057
MDCVRP-171	5.38%	19	13	11534.90	176
MDCVRP-174	6.13%	19	231	10373.74	2786
MDCVRP-236	4.86%	12	13	8501.00	173
MDCVRP-250	0.19%	13	3	9796.07	15
MDCVRP-302	3.36%	25	13	9771.93	604
MDCVRP-326	0.12%	45	5	18741.80	68
MDCVRP-327	5.98%	16	91	21225.95	4246
MDCVRP-339	8.92%	19	45	22367.17	466

Instance	Rg (%)	Rt (s)	Nodes	OPT	t (s)
MDCVRP-387	3.23%	37	41	19946.78	606
MDCVRP-390	3.39%	31	19	19986.70	242
MDCVRP-395	3.14%	24	43	21249.03	1298
MDCVRP-421	5.79%	13	29	21224.26	257
MDCVRP-423	5.78%	14	21	21245.60	401
MDCVRP-425	4.60%	28	49	18710.86	1371
MDCVRP-439	5.86%	16	29	21190.06	602
MDCVRP-464	5.86%	15	53	21199.69	1042
MDCVRP-470	0.11%	45	3	18730.92	66
MDCVRP-488	0.80%	21	13	20116.83	917
MDCVRP-502	0.52%	24	27	20104.00	1116
MDCVRP-587	5.87%	12	29	21238.92	934
MDCVRP-613	3.05%	24	19	21311.24	1960
MDCVRP-622	7.59%	29	139	11037.13	2514
MDCVRP-632	3.59%	11	9	9146.32	400
MDCVRP-639	8.95%	33	111	10380.29	849
MDCVRP-641	2.52%	38	25	10480.34	2146
MDCVRP-658	8.70%	14	15	7171.93	386
MDCVRP-685	4.98%	25	69	11024.04	1086
MDCVRP-707	6.46%	18	19	9730.32	282
MDCVRP-734	3.26%	32	9	9755.94	472
MDCVRP-737	5.88%	13	63	7224.22	1342
MDCVRP-738	0.48%	24	357	7225.52	6935
MDCVRP-739	0.10%	68	7	7213.34	126
MDCVRP-750	5.30%	26	53	11050.04	1062
MDCVRP-756	7.79%	34	199	10360.89	1766
MDCVRP-803	0.11%	33	5	17427.52	303
MDCVRP-822	0.56%	53	17	6514.35	619
MDCVRP-1000	0.37%	41	9	9039.34	175
MDCVRP-1002	8.44%	35	273	10943.38	2968
MDCVRP-1025	6.69%	29	113	11452.94	1338
MDCVRP-1041	7.29%	21	61	11565.85	826
MDCVRP-1165	8.03%	36	187	10846.28	4080
MDCVRP-1172	0.23%	43	17	8401.45	1185
MDCVRP-1238	0.36%	37	453	9021.62	7184
MDCVRP-1253	7.94%	38	187	21046.28	3273
MDCVRP-1262	0.24%	44	53	17418.74	787
MDCVRP-1268	3.45%	50	3	17335.24	57
MDCVRP-1279	0.62%	48	37	18770.68	642
MDCVRP-1284	8.66%	34	373	21010.83	3452
MDCVRP-1286	4.14%	52	3	17335.31	71
MDCVRP-1293	0.17%	50	19	18638.32	221
MDCVRP-1299	1.88%	23	75	18548.08	1352
MDCVRP-1300	3.02%	32	79	19840.65	4078
MDCVRP-1305	2.54%	18	81	17356.16	1835
MDCVRP-1318	3.25%	31	3	18526.70	45
MDCVRP-1359	3.85%	29	9	21081.58	174
MDCVRP-1364	5.37%	19	7	20990.38	85
MDCVRP-1368	3.47%	34	3	17407.77	41
MDCVRP-1370	3.14%	35	3	18664.96	42
MDCVRP-1411	2.22%	36	9	19790.23	176
MDCVRP-1465	7.66%	31	29	20987.81	1380
MDCVRP-1489	5.09%	23	25	21047.19	328
MDCVRP-1582	0.56%	40	29	7087.31	1007
MDCVRP-1612	5.66%	14	3	7034.78	40
MDCVRP-1614	0.25%	70	25	6451.33	536
MDCVRP-1617	4.26%	21	23	7085.51	343
MDCVRP-1663	4.26%	11	9	9661.81	425
MDCVRP-1687	0.42%	39	101	7833.00	2030
MDCVRP-1738	3.97%	12	3	14966.96	21
MDCVRP-1739	2.57%	13	15	16164.03	267
MDCVRP-1765	0.22%	22	51	18555.84	937
MDCVRP-1799	2.72%	27	75	13597.45	1282
MDCVRP-1832	1.88%	36	79	9069.13	885
MDCVRP-1850	0.17%	37	67	11711.82	640
MDCVRP-1908	3.28%	11	7	18673.75	62
MDCVRP-1910	0.16%	44	77	17543.72	994
MDCVRP-1933	2.22%	13	3	17343.24	24
MDCVRP-1963	7.95%	31	27	11507.00	393

Instance	Rg (%)	Rt (s)	Nodes	OPT	t (s)
MDCVRP-2014	1.03%	32	21	11015.82	242
MDCVRP-2035	2.90%	16	185	11039.82	1333
MDCVRP-2055	3.26%	37	15	9580.49	776
MDCVRP-2084	0.40%	43	21	9135.65	338
MDCVRP-2095	0.68%	51	21	8987.88	261
MDCVRP-2136	0.98%	23	81	9109.73	2897
MDCVRP-2253	6.47%	10	7	18608.38	82
MDCVRP-2315	5.32%	13	9	11000.87	122
MDCVRP-2317	0.50%	65	5	10443.01	145
MDCVRP-2336	0.40%	68	37	10434.83	790
MDCVRP-2347	6.88%	18	89	12266.93	1361
MDCVRP-2355	0.17%	78	5	11181.37	172
MDCVRP-2423	7.26%	19	727	12134.41	5752
MDCVRP-2436	2.90%	23	161	10898.42	4983
MDCVRP-2448	2.45%	20	53	10947.31	1836
MDCVRP-2456	3.39%	49	7	8911.51	294
MDCVRP-2459	4.24%	25	189	10815.62	2751
MDCVRP-2460	7.54%	30	111	11394.43	3076
MDCVRP-2463	3.62%	34	19	11443.10	188
MDCVRP-2485	4.35%	21	13	10914.65	691
MDCVRP-2490	4.47%	30	5	10900.54	86
MDCVRP-2538	3.30%	33	251	11528.30	4971
MDCVRP-2543	0.51%	40	181	7860.02	4444
MDCVRP-2556	2.22%	25	81	10936.53	2353
MDCVRP-2565	2.57%	25	17	10978.89	189
MDCVRP-2594	2.11%	20	27	10560.02	764
MDCVRP-2608	2.78%	27	13	10894.13	364
MDCVRP-2621	2.32%	22	247	21173.52	8729
MDCVRP-2636	5.43%	27	37	19835.73	430
MDCVRP-2641	1.87%	28	3	21073.55	35
MDCVRP-2651	3.52%	30	9	17369.17	153
MDCVRP-2687	0.20%	31	281	21133.94	6823
MDCVRP-2726	0.31%	61	7	13837.23	195
MDCVRP-2767	5.96%	31	17	10222.75	334
MDCVRP-2769	0.16%	71	5	6488.16	204
MDCVRP-2771	3.21%	16	23	12821.29	425
MDCVRP-2784	1.32%	35	41	10446.79	892
MDCVRP-2805	6.10%	11	17	9661.41	255
MDCVRP-2814	1.40%	31	87	10506.75	3394
MDCVRP-2880	6.93%	29	37	12210.07	537
MDCVRP-2898	2.79%	33	9	10977.58	200
MDCVRP-2946	0.08%	87	5	14977.40	178
MDCVRP-2957	7.97%	27	13	22349.56	470
MDCVRP-2970	3.45%	40	3	17419.64	68
MDCVRP-2984	0.09%	27	3	17459.90	338
MDCVRP-3006	0.09%	21	11	15067.69	211
MDCVRP-3025	4.50%	11	39	18645.81	964
MDCVRP-3035	0.19%	32	63	20058.83	737
MDCVRP-3040	4.12%	12	9	14953.18	187
MDCVRP-3076	4.38%	22	5	10950.35	49
MDCVRP-3080	4.97%	39	45	10319.53	379
MDCVRP-3089	5.81%	18	19	7113.93	379
MDCVRP-3093	1.92%	38	33	12450.04	637
MDCVRP-3109	2.68%	12	239	9717.28	2477
MDCVRP-3128	8.03%	12	25	7711.85	272
MDCVRP-3141	7.84%	18	9	7782.68	188
MDCVRP-3147	8.21%	15	11	7175.18	159
MDCVRP-3151	5.60%	16	21	7171.69	365
MDCVRP-3160	0.18%	72	5	6538.21	193
MDCVRP-3162	0.55%	38	265	7167.42	6171
MDCVRP-3174	4.68%	17	59	7166.31	1448
MDCVRP-3185	0.78%	60	199	9799.32	2507
MDCVRP-3196	8.14%	17	177	7698.86	3700
MDCVRP-3237	6.16%	42	37	10195.02	885
MDCVRP-3240	3.29%	36	121	11507.56	2298
MDCVRP-3274	4.50%	27	25	11536.16	1548
MDCVRP-3275	1.94%	42	19	10956.41	1189
MDCVRP-3324	0.32%	20	19	7832.43	709
MDCVRP-3326	0.35%	49	13	9722.02	171

Instance	Rg (%)	Rt (s)	Nodes	OPT	t (s)
MDCVRP-3350	6.10%	28	49	19920.29	482
MDCVRP-3354	7.16%	24	13	22365.66	281
MDCVRP-3357	0.28%	34	55	21226.58	761
MDCVRP-3360	2.75%	33	143	18772.84	8454
MDCVRP-3367	7.24%	14	85	16161.18	1558
MDCVRP-3385	7.30%	13	7	16154.08	74
MDCVRP-3397	4.37%	25	23	21227.42	391
MDCVRP-3403	2.90%	35	27	21109.44	876
MDCVRP-3413	7.57%	23	29	23426.59	409
MDCVRP-3422	0.14%	28	19	20051.45	209
MDCVRP-3430	0.14%	29	73	17329.23	936
MDCVRP-3431	0.26%	67	149	17454.82	2463
MDCVRP-3437	5.03%	12	7	16118.20	87
MDCVRP-3448	2.95%	28	85	22292.76	1687
MDCVRP-3453	3.53%	12	5	17386.94	55
MDCVRP-3508	0.98%	20	23	10377.62	1179
MDCVRP-3518	0.00%	40	1	7088.59	39
MDCVRP-3526	2.95%	19	5	10260.44	178
MDCVRP-3534	0.23%	113	7	6530.05	203
MDCVRP-3537	0.26%	67	3	6495.48	96
MDCVRP-3544	2.89%	28	3	10238.41	42
MDCVRP-3608	4.35%	16	7	7202.53	108
MDCVRP-3643	4.47%	18	9	7182.04	184
MDCVRP-3660	0.22%	26	3	7109.64	64

Table 14 contains instances for which the initial primal bound has not been proven optimal before these experiments.

Table 14: Detailed results for algorithm BCP_{best} for 16 instances of the VRP-CMD with unknown optimal solutions

Instance	IPB	Rg (%)	Rt (s)	Nodes	Fg (%)	BPB	t (s)
MDCVRP-87	10521.07	10.48%	28	2563	0.87%	10410.72	108000
MDCVRP-235	11144.42	4.44%	22	473	0.00%	11112.75	34508
MDCVRP-453	21285.26	3.36%	23	1749	0.28%	21279.71	108000
MDCVRP-487	21352.94	3.62%	24	203	0.00%	21255.48	12772
MDCVRP-528	18679.39	4.46%	19	629	0.00%	18679.39	14241
MDCVRP-582	23660.82	8.59%	18	575	0.00%	23660.82	17636
MDCVRP-599	23579.35	8.53%	25	2639	0.25%	23578.14	108000
MDCVRP-612	23708.73	8.65%	20	3591	0.17%	23708.73	108000
MDCVRP-653	9200.66	3.02%	60	491	0.00%	9200.59	37617
MDCVRP-1280	23567.65	10.47%	23	5913	0.36%	23520.74	108000
MDCVRP-1541	10231.13	6.42%	47	1075	0.00%	10231.13	12851
MDCVRP-2313	11069.50	2.16%	16	1267	0.00%	11069.50	13915
MDCVRP-2431	11551.42	4.10%	35	837	0.00%	11470.91	49199
MDCVRP-2444	10288.95	8.57%	13	2381	4.19%	10288.95	108000
MDCVRP-2734	22379.58	3.74%	20	5899	0.00%	22372.04	96868
MDCVRP-2745	19920.04	5.41%	26	6587	0.39%	19920.04	108000

D Detailed results for VRPTW with Shifts

In the following tables, BKS stands for the best-known solution value in the literature, RDB for the root dual bound, Rt for the time spent in the root node, Nodes for the number of nodes explored in the branch-and-bound tree, BDB for the best found dual bound, BPB for the best primal bound, and t for the total time. Improved solution values in comparison with the

literature are in bold. If the best bounds are equal then the instances are solved to optimality. Sometimes the total solution time is smaller than the time limit of 30 hours, but the best dual and primal bounds are not equal. In this case, the algorithm was interrupted due to a very large solution time of the labeling algorithm.

Table 15: Detailed results for algorithm BCP_{all} without initial upper bounds on instances of the VRPTW-S with small shift capacities ($\rho = 1.05$)

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
A-25-c101	287.2	283.2	11	3	287.2	287.2	16
A-25-c102	263.8	263.8	9	1	263.8	263.8	9
A-25-c103	251.8	251.8	6	1	251.8	251.8	5
A-25-c104	249.2	249.2	22	1	249.2	249.2	22
A-25-c105	265.0	263.5	4	3	265.0	265.0	7
A-25-c106	287.2	282.8	12	3	287.2	287.2	18
A-25-c107	265.0	263.2	9	3	265.0	265.0	14
A-25-c108	263.2	261.5	9	3	263.2	263.2	19
A-25-c109	256.5	256.5	11	1	256.5	256.5	11
A-25-c201	288.9	288.9	6	1	288.9	288.9	6
A-25-c202	281.7	281.7	3	1	281.7	281.7	3
A-25-c203	279.3	279.3	3	1	279.3	279.3	3
A-25-c204	279.3	279.3	5	1	279.3	279.3	5
A-25-c205	288.9	288.9	3	1	288.9	288.9	3
A-25-c206	286.7	286.7	2	1	286.7	286.7	1
A-25-c207	284.0	284.0	3	1	284.0	284.0	3
A-25-c208	285.9	285.9	4	1	285.9	285.9	4
A-25-r101	710.9	710.9	0	1	710.9	710.9	0
A-25-r102	580.7	577.1	3	3	580.7	580.7	5
A-25-r103	504.7	496.5	5	3	504.7	504.7	29
A-25-r104	483.6	475.9	4	3	483.6	483.6	10
A-25-r105	596.8	596.8	2	1	596.8	596.8	2
A-25-r106	490.4	490.4	5	1	490.4	490.4	5
A-25-r107	457.2	454.0	6	3	457.2	457.2	11
A-25-r108	437.7	437.7	7	1	437.7	437.7	7
A-25-r109	497.3	492.7	4	3	497.3	497.3	6
A-25-r110	467.1	458.7	7	3	467.1	467.1	11
A-25-r111	471.7	467.0	6	3	471.7	471.7	11
A-25-r112	427.9	420.8	9	3	427.9	427.9	19
A-25-r201	502.1	502.1	6	1	502.1	502.1	6
A-25-r202	437.8	435.8	7	3	437.8	437.8	17
A-25-r203	404.0	404.0	3	1	404.0	404.0	3
A-25-r204	375.4	375.4	2	1	375.4	375.4	2
A-25-r205	421.2	417.4	9	3	421.2	421.2	17
A-25-r206	384.2	384.2	2	1	384.2	384.2	2
A-25-r207	375.4	375.4	2	1	375.4	375.4	2
A-25-r208	367.4	367.4	4	1	367.4	367.4	4
A-25-r209	383.7	383.7	2	1	383.7	383.7	2
A-25-r210	421.5	421.5	3	1	421.5	421.5	3
A-25-r211	364.9	364.9	2	1	364.9	364.9	2
A-25-rc101	608.7	600.4	1	1	600.4	**600.4**	1
A-25-rc102	470.4	454.8	11	3	470.4	470.4	17
A-25-rc103	427.8	427.8	15	1	427.8	427.8	15
A-25-rc104	410.3	397.7	32	1	397.7	**397.7**	32
A-25-rc105	535.6	534.7	4	3	535.6	535.6	6
A-25-rc106	472.9	472.9	2	1	472.9	472.9	2
A-25-rc107	434.8	434.8	4	1	434.8	434.8	4
A-25-rc108	∞	383.7	41	3	397.0	**397.0**	81
A-25-rc201	496.4	478.5	11	1	478.5	**478.5**	11
A-25-rc202	381.0	381.0	13	1	381.0	381.0	13
A-25-rc203	367.1	361.9	18	1	361.9	**361.9**	18
A-25-rc204	∞	334.7	85	1	334.7	**334.7**	84
A-25-rc205	376.3	376.3	9	1	376.3	376.3	9
A-25-rc206	369.5	369.5	8	1	369.5	369.5	7
A-25-rc207	337.2	337.2	83	1	337.2	337.2	83
A-25-rc208	∞	329.5	254	1	329.5	**329.5**	254
A-50-c101	499.5	487.4	12	3	499.5	499.5	20
A-50-c102	467.5	464.0	9	3	467.5	467.5	19

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
A-50-c103	∞	457.5	12	3	461.4	**461.4**	39
A-50-c104	∞	419.0	113	1	419.0	**419.0**	112
A-50-c105	466.6	462.3	7	3	466.6	466.6	14
A-50-c106	499.5	476.4	9	3	499.5	499.5	34
A-50-c107	464.8	458.9	6	3	464.8	464.8	15
A-50-c108	500.3	456.9	11	3	462.4	**462.4**	21
A-50-c109	∞	445.3	17	5	454.7	**454.7**	57
A-50-c201	429.8	429.8	6	1	429.8	429.8	6
A-50-c202	421.5	421.5	59	1	421.5	421.5	59
A-50-c203	∞	414.1	79	1	414.1	**414.1**	79
A-50-c204	∞	402.5	484	3	403.2	**403.2**	1544
A-50-c205	427.8	427.8	12	1	427.8	427.8	12
A-50-c206	427.8	427.8	15	1	427.8	427.8	15
A-50-c207	∞	425.7	55	3	426.1	**426.1**	101
A-50-c208	424.2	422.5	35	3	422.7	**422.7**	65
A-50-r101	1217.3	1209.8	4	3	1217.3	1217.3	6
A-50-r102	980.6	975.8	4	3	980.6	980.6	10
A-50-r103	820.3	812.3	10	3	820.3	820.3	49
A-50-r104	664.6	662.2	16	3	664.6	664.6	43
A-50-r105	1037.9	1036.2	5	3	1037.9	1037.9	11
A-50-r106	875.6	870.5	6	3	875.6	875.6	16
A-50-r107	777.0	763.1	8	7	773.8	**773.8**	104
A-50-r108	644.8	638.1	55	3	644.8	644.8	140
A-50-r109	891.3	838.8	8	3	852.3	**852.3**	58
A-50-r110	749.0	741.4	14	3	749.0	749.0	38
A-50-r111	752.4	749.4	14	3	752.4	752.4	28
A-50-r112	665.1	659.2	48	3	665.1	665.1	123
A-50-r201	∞	834.2	10	7	844.1	**844.1**	95
A-50-r202	732.7	729.6	37	3	732.7	732.7	80
A-50-r203	632.3	627.6	92	3	632.3	632.3	258
A-50-r204	∞	528.3	356	3	530.0	**530.0**	1208
A-50-r205	716.4	713.4	44	3	716.4	716.4	86
A-50-r206	649.6	648.6	104	3	649.6	649.6	334
A-50-r207	587.8	587.5	337	3	587.8	587.8	1059
A-50-r208	∞	502.4	53	1	502.4	**502.4**	53
A-50-r209	619.1	616.2	94	3	619.1	619.1	182
A-50-r210	657.0	657.0	193	1	657.0	657.0	193
A-50-r211	541.5	541.5	125	1	541.5	541.5	124
A-50-rc101	1174.1	1131.1	7	3	1136.8	**1136.8**	8
A-50-rc102	975.8	951.7	23	3	973.6	**973.6**	39
A-50-rc103	∞	824.8	38	3	836.5	**836.5**	91
A-50-rc104	∞	669.7	110	3	697.4	**697.4**	520
A-50-rc105	∞	1043.6	40	1	1043.6	**1043.6**	45
A-50-rc106	∞	939.5	50	3	956.1	**956.1**	90
A-50-rc107	∞	766.2	501	1	766.2	**766.2**	501
A-50-rc108	∞	664.1	272	21	724.3	**724.3**	4640
A-50-rc201	∞	809.6	51	3	828.6	**828.6**	122
A-50-rc202	∞	711.2	242	1	711.2	**711.2**	242
A-50-rc203	∞	645.2	346	3	654.4	**654.4**	883
A-50-rc204	∞	497.7	377	13	533.3	**533.3**	4390
A-50-rc205	717.7	717.7	53	1	717.7	717.7	53
A-50-rc206	∞	662.9	135	3	669.1	**669.1**	205
A-50-rc207	∞	588.1	306	21	617.0	**617.0**	6180
A-50-rc208	∞	482.3	336	17	510.4	**510.4**	6503
A-100-c101	1114.5	1109.1	23	3	1114.5	1114.5	40
A-100-c102	1036.3	1033.3	41	3	1036.3	1036.3	77
A-100-c103	∞	1010.0	104	1	1010.0	**1010.0**	104
A-100-c104	∞	935.4	552	5	943.6	**943.6**	4518
A-100-c105	1071.6	1063.8	18	3	1071.6	1071.6	43
A-100-c106	∞	1072.5	27	3	1078.9	**1078.9**	61
A-100-c107	1030.6	1030.6	14	1	1030.6	1030.6	13
A-100-c108	1029.3	1029.3	25	1	1029.3	1029.3	32
A-100-c109	∞	977.3	49	7	995.1	**995.1**	287
A-100-c201	∞	675.2	113	3	683.6	**683.6**	341
A-100-c202	∞	672.9	348	3	678.5	**678.5**	737
A-100-c203	∞	663.0	710	3	669.1	**669.1**	1383
A-100-c204	∞	630.8	864	1	630.8	∞	6578
A-100-c205	∞	660.8	125	5	674.1	**674.1**	479
A-100-c206	∞	656.6	171	3	664.1	**664.1**	349

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
A-100-c207	∞	653.1	255	3	663.9	**663.9**	743
A-100-c208	∞	649.8	488	3	660.9	**660.9**	967
A-100-r101	1840.6	1832.4	11	3	1840.6	1840.6	38
A-100-r102	1535.4	1534.2	17	3	1535.4	1535.4	40
A-100-r103	∞	1222.9	29	3	1223.7	**1223.7**	86
A-100-r104	∞	1001.2	197	27	1009.0	**1009.0**	5793
A-100-r105	1465.2	1459.4	20	3	1465.2	1465.2	72
A-100-r106	1294.9	1286.5	41	27	1294.9	1294.9	855
A-100-r107	∞	1097.7	66	31	1106.1	**1106.1**	3544
A-100-r108	∞	955.2	241	421	968.5	**968.5**	66955
A-100-r109	∞	1225.9	54	17	1237.8	**1237.8**	1002
A-100-r110	∞	1120.8	141	7	1126.8	**1126.8**	597
A-100-r111	∞	1093.0	90	13	1100.6	**1100.6**	996
A-100-r112	∞	982.9	400	11	987.2	**987.2**	3741
A-100-r201	1188.5	1184.4	163	7	1188.5	1188.5	694
A-100-r202	∞	1039.6	386	21	1044.2	**1044.2**	7738
A-100-r203	∞	882.3	1282	35	889.9	**889.9**	19393
A-100-r204	∞	724.5	970	173	728.4	**775.5**	108002
A-100-r205	∞	962.3	435	33	973.0	**973.0**	11826
A-100-r206	∞	876.3	638	217	879.1	**933.5**	108004
A-100-r207	∞	793.1	690	1	793.1	∞	2640
A-100-r208	∞	690.6	1041	1	690.6	∞	3303
A-100-r209	∞	855.3	534	83	871.5	**871.5**	34124
A-100-r210	∞	896.8	548	137	904.5	**904.5**	63478
A-100-r211	∞	735.7	618	39	738.1	∞	26075
A-100-rc101	1877.3	1847.1	29	3	1853.0	**1853.0**	83
A-100-rc102	1598.6	1581.6	84	11	1596.0	**1596.0**	453
A-100-rc103	∞	1349.4	142	11	1362.7	**1362.7**	1290
A-100-rc104	∞	1181.8	262	9	1205.4	**1205.4**	2640
A-100-rc105	1720.7	1705.8	63	3	1712.5	**1712.5**	139
A-100-rc106	∞	1513.3	81	9	1518.8	**1518.8**	334
A-100-rc107	∞	1310.7	93	11	1321.9	**1321.9**	712
A-100-rc108	∞	1186.8	280	305	1209.5	**1209.5**	78481
A-100-rc201	∞	1290.6	79	9	1298.7	**1298.7**	641
A-100-rc202	∞	1111.7	205	9	1116.4	**1116.4**	1791
A-100-rc203	∞	939.6	952	3	944.7	**944.7**	1873
A-100-rc204	∞	776.9	1583	3	778.7	∞	5049
A-100-rc205	∞	1167.1	225	19	1176.4	**1176.4**	3132
A-100-rc206	∞	1062.4	309	15	1072.7	**1072.7**	3349
A-100-rc207	∞	956.6	798	95	970.9	**970.9**	35039
A-100-rc208	∞	766.2	2496	211	769.4	**812.3**	108014

Table 16: Detailed results for algorithm BCP_{all} without initial upper bounds on instances of the VRPTW-S with medium shift capacities ($\rho = 1.2$)

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
B-25-c101	265.2	265.2	10	1	265.2	265.2	10
B-25-c102	251.6	251.6	10	1	251.6	251.6	10
B-25-c103	247.4	244.4	24	3	247.4	247.4	38
B-25-c104	∞	241.9	56	1	241.9	**241.9**	56
B-25-c105	258.8	258.8	8	1	258.8	258.8	8
B-25-c106	265.2	265.2	12	1	265.2	265.2	12
B-25-c107	256.6	256.6	6	1	256.6	256.6	6
B-25-c108	256.6	256.6	26	3	256.6	256.6	43
B-25-c109	∞	252.8	75	3	254.8	**254.8**	137
B-25-c201	279.6	279.6	2	1	279.6	279.6	2
B-25-c202	272.9	272.9	3	1	272.9	272.9	3
B-25-c203	269.9	269.9	3	1	269.9	269.9	3
B-25-c204	269.9	269.9	6	1	269.9	269.9	6
B-25-c205	279.6	279.6	3	1	279.6	279.6	3
B-25-c206	277.4	277.4	2	1	277.4	277.4	2
B-25-c207	274.5	274.5	4	1	274.5	274.5	4
B-25-c208	276.6	276.6	3	1	276.6	276.6	3
B-25-r101	671.9	671.9	0	1	671.9	671.9	0

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
B-25-r102	557.4	556.1	3	3	557.4	557.4	5
B-25-r103	478.2	476.5	3	3	478.2	478.2	7
B-25-r104	451.8	451.8	4	1	451.8	451.8	4
B-25-r105	566.4	565.6	1	1	565.6	**565.6**	1
B-25-r106	483.2	474.4	6	3	483.2	483.2	9
B-25-r107	443.3	443.3	3	1	443.3	443.3	3
B-25-r108	426.9	426.9	4	1	426.9	426.9	4
B-25-r109	481.7	472.6	4	3	481.7	481.7	6
B-25-r110	451.9	448.1	5	3	451.9	451.9	9
B-25-r111	461.3	455.5	4	3	461.3	461.3	9
B-25-r112	405.9	405.9	5	1	405.9	405.9	5
B-25-r201	496.4	491.3	6	3	496.4	496.4	12
B-25-r202	429.4	423.4	4	3	429.4	429.4	10
B-25-r203	395.9	395.9	3	1	395.9	395.9	3
B-25-r204	368.3	368.3	4	1	368.3	368.3	4
B-25-r205	402.6	402.6	1	1	402.6	402.6	1
B-25-r206	382.0	382.0	4	1	382.0	382.0	3
B-25-r207	363.1	363.1	1	1	363.1	363.1	1
B-25-r208	356.5	356.5	1	1	356.5	356.5	1
B-25-r209	376.4	376.4	1	1	376.4	376.4	1
B-25-r210	413.1	413.1	3	1	413.1	413.1	3
B-25-r211	361.6	361.6	3	1	361.6	361.6	3
B-25-rc101	534.9	534.9	1	1	534.9	534.9	1
B-25-rc102	417.7	417.7	3	1	417.7	417.7	3
B-25-rc103	398.7	398.7	6	1	398.7	398.7	6
B-25-rc104	368.3	368.3	11	1	368.3	368.3	11
B-25-rc105	483.7	479.1	2	3	483.7	483.7	4
B-25-rc106	467.8	456.4	5	3	460.7	**460.7**	7
B-25-rc107	∞	409.5	24	3	430.0	**430.0**	37
B-25-rc108	369.8	363.4	27	1	363.4	**363.4**	27
B-25-rc201	446.6	429.2	6	1	429.2	**429.2**	6
B-25-rc202	338.0	338.0	1	1	338.0	338.0	1
B-25-rc203	326.9	326.9	2	1	326.9	326.9	2
B-25-rc204	299.7	299.7	2	1	299.7	299.7	2
B-25-rc205	338.0	338.0	1	1	338.0	338.0	1
B-25-rc206	334.4	334.4	1	1	334.4	334.4	1
B-25-rc207	298.3	298.3	1	1	298.3	298.3	1
B-25-rc208	294.5	294.5	3	1	294.5	294.5	3
B-50-c101	455.1	431.3	5	1	431.3	**431.3**	5
B-50-c102	442.1	425.8	11	3	428.3	**428.3**	25
B-50-c103	424.3	422.9	13	3	424.3	424.3	34
B-50-c104	∞	395.0	260	9	409.0	**409.0**	3362
B-50-c105	428.9	424.4	7	3	428.9	428.9	16
B-50-c106	431.3	431.3	6	1	431.3	431.3	6
B-50-c107	425.7	423.7	10	3	425.7	425.7	22
B-50-c108	424.5	421.3	8	3	424.5	424.5	20
B-50-c109	∞	416.7	25	3	421.8	**421.8**	49
B-50-c201	412.3	412.3	4	1	412.3	412.3	4
B-50-c202	407.2	406.7	59	3	407.2	407.2	95
B-50-c203	400.0	396.7	49	1	396.7	**396.7**	49
B-50-c204	∞	390.3	144	1	390.3	**390.3**	144
B-50-c205	411.9	411.9	5	1	411.9	411.9	5
B-50-c206	411.9	411.9	12	1	411.9	411.9	12
B-50-c207	∞	408.3	72	3	411.7	**411.7**	131
B-50-c208	402.6	402.6	8	1	402.6	402.6	8
B-50-r101	1136.4	1131.6	4	3	1136.4	1136.4	5
B-50-r102	933.7	930.7	4	3	933.7	933.7	9
B-50-r103	792.9	787.0	10	3	792.9	792.9	24
B-50-r104	650.0	642.0	39	3	650.0	650.0	123
B-50-r105	982.0	973.2	4	3	979.1	**979.1**	10
B-50-r106	841.0	836.7	7	3	841.0	841.0	17
B-50-r107	741.7	737.0	16	3	741.7	741.7	38
B-50-r108	∞	620.7	101	3	628.2	**628.2**	321
B-50-r109	817.9	813.5	10	3	817.9	817.9	19
B-50-r110	733.9	725.7	13	3	733.9	733.9	32
B-50-r111	733.1	730.5	20	3	733.1	733.1	41
B-50-r112	649.4	646.7	45	3	649.4	649.4	86
B-50-r201	819.7	815.6	12	3	819.7	819.7	25
B-50-r202	730.6	717.5	40	3	720.6	**720.6**	76

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
B-50-r203	619.3	616.9	79	3	619.3	619.3	177
B-50-r204	∞	520.9	220	3	521.1	**521.1**	445
B-50-r205	705.0	698.8	42	5	705.0	705.0	173
B-50-r206	663.4	643.2	135	3	647.3	**647.3**	337
B-50-r207	582.7	582.7	322	1	582.7	582.7	322
B-50-r208	∞	498.5	60	1	498.5	**498.5**	60
B-50-r209	611.3	609.0	55	3	611.3	611.3	126
B-50-r210	656.6	653.4	176	3	656.6	656.6	435
B-50-r211	536.9	536.9	139	1	536.9	536.9	139
B-50-rc101	1098.6	1034.4	6	3	1051.1	**1051.1**	15
B-50-rc102	912.1	909.0	31	3	910.0	**910.0**	49
B-50-rc103	∞	768.7	54	3	783.3	**783.3**	96
B-50-rc104	∞	628.0	124	3	631.8	**631.8**	361
B-50-rc105	∞	964.6	45	3	965.8	**965.8**	62
B-50-rc106	∞	866.7	45	3	886.7	**886.7**	87
B-50-rc107	∞	718.8	359	3	726.7	**726.7**	824
B-50-rc108	∞	624.5	170	11	700.7	**700.7**	2351
B-50-rc201	809.8	764.4	42	3	782.2	**782.2**	76
B-50-rc202	690.7	669.4	40	3	672.3	**672.3**	73
B-50-rc203	∞	610.4	72	1	610.4	**610.4**	72
B-50-rc204	∞	476.1	411	17	504.7	**504.7**	5281
B-50-rc205	680.3	680.3	15	1	680.3	680.3	15
B-50-rc206	636.0	614.5	34	5	636.0	636.0	227
B-50-rc207	∞	561.8	146	37	582.5	**582.5**	5441
B-50-rc208	∞	469.2	383	157	496.7	**496.7**	48087
B-100-c101	991.8	984.9	27	3	991.8	991.8	67
B-100-c102	∞	954.2	42	3	955.2	**955.2**	97
B-100-c103	∞	942.9	149	9	949.2	**949.2**	506
B-100-c104	∞	897.1	359	5	904.7	**904.7**	4432
B-100-c105	964.6	961.4	21	3	964.6	964.6	44
B-100-c106	972.4	967.1	36	3	972.4	972.4	70
B-100-c107	954.6	949.6	20	3	954.6	954.6	40
B-100-c108	∞	948.6	32	3	954.5	**954.5**	62
B-100-c109	∞	909.6	46	9	911.2	**911.2**	277
B-100-c201	∞	662.9	278	3	671.7	**671.7**	462
B-100-c202	∞	656.4	461	5	667.6	**667.6**	1188
B-100-c203	∞	645.2	900	9	655.6	**655.6**	4088
B-100-c204	∞	624.4	1573	1	624.4	**681.3**	3443
B-100-c205	∞	653.8	233	9	668.6	**668.6**	1088
B-100-c206	∞	643.8	256	11	664.1	**664.1**	1339
B-100-c207	∞	639.6	308	3	647.2	**647.2**	616
B-100-c208	∞	633.8	388	13	647.9	**647.9**	2632
B-100-r101	1721.6	1708.3	14	3	1715.7	**1715.7**	33
B-100-r102	1499.5	1498.6	18	3	1499.5	1499.5	45
B-100-r103	∞	1199.4	42	3	1201.5	**1201.5**	108
B-100-r104	1391.3	978.0	229	33	985.8	**985.8**	6548
B-100-r105	∞	1390.2	22	3	1391.3	**1391.3**	42
B-100-r106	∞	1260.3	43	9	1266.1	**1266.1**	415
B-100-r107	∞	1071.5	96	11	1077.6	**1077.6**	1348
B-100-r108	∞	931.6	282	9	942.0	**942.0**	2686
B-100-r109	∞	1188.0	64	37	1199.1	**1199.1**	2054
B-100-r110	∞	1091.7	153	7	1095.3	**1095.3**	675
B-100-r111	∞	1065.1	75	3	1065.5	**1065.5**	136
B-100-r112	∞	964.4	398	57	975.7	**975.7**	16094
B-100-r201	1167.7	1164.2	98	5	1167.7	1167.7	269
B-100-r202	∞	1031.9	460	5	1035.6	**1035.6**	2381
B-100-r203	∞	869.5	632	63	879.2	**879.2**	32406
B-100-r204	∞	723.5	1099	137	727.4	**796.8**	108022
B-100-r205	∞	954.4	530	59	965.0	**965.0**	21217
B-100-r206	∞	868.3	655	265	874.2	**892.7**	108004
B-100-r207	∞	790.7	1023	3	790.7	∞	5294
B-100-r208	∞	689.5	1306	7	691.5	∞	8418
B-100-r209	∞	851.2	718	183	853.4	∞	108012
B-100-r210	∞	893.1	583	13	900.5	**900.5**	6977
B-100-r211	∞	735.0	834	1	735.0	∞	2701
B-100-rc101	1759.4	1730.1	31	3	1738.2	**1738.2**	87
B-100-rc102	1532.9	1513.5	112	5	1520.8	**1520.8**	295
B-100-rc103	∞	1294.8	171	3	1302.1	**1302.1**	395
B-100-rc104	∞	1143.4	262	21	1158.2	**1158.2**	3695

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
B-100-rc105	1662.1	1614.0	61	3	1617.8	**1617.8**	131
B-100-rc106	∞	1435.7	79	7	1442.9	**1442.9**	268
B-100-rc107	∞	1263.7	97	9	1274.7	**1274.7**	717
B-100-rc108	∞	1152.9	285	15	1166.3	**1166.3**	3922
B-100-rc201	1279.8	1275.8	82	3	1279.8	1279.8	149
B-100-rc202	1116.5	1101.1	238	13	1110.4	**1110.4**	1863
B-100-rc203	∞	926.7	707	1	926.7	**974.9**	1947
B-100-rc204	∞	771.8	2407	3	773.3	∞	6659
B-100-rc205	∞	1156.8	268	3	1157.6	**1157.6**	405
B-100-rc206	∞	1050.9	287	3	1059.1	**1059.1**	590
B-100-rc207	∞	953.4	700	59	968.6	**968.6**	25852
B-100-rc208	∞	763.5	2547	9	766.8	∞	13638

Table 17: Detailed results for algorithm BCP_{all} without initial upper bounds on instances of the VRPTW-S with large shift capacities ($\rho = 1.5$)

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
C-25-c101	213.4	208.0	8	3	213.4	213.4	13
C-25-c102	212.4	207.0	15	3	212.4	212.4	27
C-25-c103	207.5	206.5	15	3	207.5	207.5	25
C-25-c104	204.4	200.9	11	3	204.4	204.4	24
C-25-c105	213.4	206.9	3	3	213.4	213.4	8
C-25-c106	213.4	208.0	11	3	213.4	213.4	17
C-25-c107	213.4	206.9	4	3	213.4	213.4	8
C-25-c108	213.4	206.9	4	3	213.4	213.4	8
C-25-c109	229.5	206.5	8	3	213.4	**213.4**	14
C-25-c201	252.1	252.1	5	1	252.1	252.1	5
C-25-c202	247.1	247.1	5	1	247.1	247.1	5
C-25-c203	244.5	244.5	6	1	244.5	244.5	6
C-25-c204	243.0	243.0	11	1	243.0	243.0	11
C-25-c205	251.3	251.3	5	1	251.3	251.3	5
C-25-c206	248.7	248.7	3	1	248.7	248.7	2
C-25-c207	245.5	245.5	7	1	245.5	245.5	7
C-25-c208	247.2	247.2	4	1	247.2	247.2	3
C-25-r101	617.1	617.1	0	1	617.1	617.1	0
C-25-r102	547.1	547.1	2	1	547.1	547.1	2
C-25-r103	454.6	454.6	1	1	454.6	454.6	1
C-25-r104	420.8	418.5	3	3	420.8	420.8	8
C-25-r105	551.9	530.7	1	1	530.7	**530.7**	1
C-25-r106	465.4	465.4	2	1	465.4	465.4	2
C-25-r107	425.8	425.8	1	1	425.8	425.8	0
C-25-r108	404.3	404.3	4	3	404.3	404.3	8
C-25-r109	447.8	445.7	2	3	447.8	447.8	5
C-25-r110	444.1	444.1	4	3	444.1	444.1	7
C-25-r111	428.8	428.8	1	1	428.8	428.8	1
C-25-r112	393.0	393.0	3	1	393.0	393.0	3
C-25-r201	467.7	465.7	4	3	467.7	467.7	9
C-25-r202	410.5	410.5	1	1	410.5	410.5	1
C-25-r203	391.4	391.4	2	1	391.4	391.4	2
C-25-r204	366.8	365.6	28	3	366.8	366.8	59
C-25-r205	402.6	400.2	6	3	402.6	402.6	14
C-25-r206	375.9	375.9	3	1	375.9	375.9	3
C-25-r207	361.6	361.6	3	1	361.6	361.6	3
C-25-r208	340.4	340.4	1	1	340.4	340.4	1
C-25-r209	376.4	375.2	8	3	376.4	376.4	22
C-25-r210	404.6	404.6	1	1	404.6	404.6	1
C-25-r211	352.0	352.0	2	1	352.0	352.0	2
C-25-rc101	462.7	462.7	1	1	462.7	462.7	1
C-25-rc102	400.8	400.8	6	1	400.8	400.8	6
C-25-rc103	389.7	388.6	28	1	388.6	**388.6**	28
C-25-rc104	361.0	361.0	30	1	361.0	361.0	30
C-25-rc105	411.3	411.3	1	1	411.3	411.3	1
C-25-rc106	396.9	396.9	2	1	396.9	396.9	2
C-25-rc107	365.2	362.1	12	3	363.8	**363.8**	19

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
C-25-rc108	∞	352.1	28	3	360.0	**360.0**	56
C-25-rc201	426.0	397.3	4	1	397.3	**397.3**	3
C-25-rc202	338.0	338.0	1	1	338.0	338.0	1
C-25-rc203	326.9	326.9	1	1	326.9	326.9	1
C-25-rc204	299.7	299.7	3	1	299.7	299.7	3
C-25-rc205	338.0	338.0	1	1	338.0	338.0	1
C-25-rc206	334.4	334.4	2	1	334.4	334.4	1
C-25-rc207	298.3	298.3	2	1	298.3	298.3	2
C-25-rc208	293.1	292.7	14	3	293.1	293.1	25
C-50-c101	400.3	377.9	10	3	384.5	**384.5**	18
C-50-c102	399.3	376.0	19	3	383.5	**383.5**	37
C-50-c103	395.0	375.6	17	3	383.5	**383.5**	36
C-50-c104	∞	369.5	27	3	378.7	**378.7**	70
C-50-c105	384.5	376.8	8	3	384.5	384.5	18
C-50-c106	384.5	376.8	9	3	384.5	384.5	18
C-50-c107	384.5	376.8	8	3	384.5	384.5	21
C-50-c108	389.3	376.7	10	3	384.5	**384.5**	26
C-50-c109	389.3	375.9	10	3	384.5	**384.5**	23
C-50-c201	395.9	392.9	45	3	395.9	395.9	104
C-50-c202	383.2	383.2	6	1	383.2	383.2	6
C-50-c203	376.9	376.9	14	1	376.9	376.9	23
C-50-c204	374.1	369.7	203	3	372.2	**372.2**	894
C-50-c205	389.4	389.4	12	1	389.4	389.4	12
C-50-c206	389.4	389.4	41	1	389.4	389.4	48
C-50-c207	377.1	377.1	13	1	377.1	377.1	13
C-50-c208	386.2	385.4	150	3	386.2	386.2	239
C-50-r101	1049.0	1049.0	3	1	1049.0	1049.0	3
C-50-r102	902.8	902.8	3	1	902.8	902.8	5
C-50-r103	760.6	757.5	7	3	760.6	760.6	17
C-50-r104	636.9	628.2	24	3	629.4	**629.4**	53
C-50-r105	926.1	906.1	5	3	906.3	**906.3**	10
C-50-r106	791.6	791.6	6	3	791.6	791.6	17
C-50-r107	720.0	705.4	16	3	708.3	**708.3**	35
C-50-r108	∞	608.5	95	3	611.1	**611.1**	303
C-50-r109	791.9	788.3	6	3	791.9	791.9	14
C-50-r110	717.3	705.2	7	1	705.2	**705.2**	11
C-50-r111	714.0	713.8	15	3	714.0	714.0	27
C-50-r112	657.2	634.7	26	3	637.1	**637.1**	52
C-50-r201	803.3	800.2	9	3	800.6	**800.6**	21
C-50-r202	765.4	708.2	26	3	713.8	**713.8**	53
C-50-r203	617.8	613.8	67	3	615.2	**615.2**	114
C-50-r204	∞	516.7	302	3	518.4	**518.4**	627
C-50-r205	690.9	690.9	13	1	690.9	690.9	13
C-50-r206	669.3	634.8	72	3	637.7	**637.7**	186
C-50-r207	∞	580.3	354	3	581.5	**581.5**	836
C-50-r208	∞	495.6	231	3	498.5	**498.5**	542
C-50-r209	600.6	600.6	10	1	600.6	600.6	10
C-50-r210	∞	648.2	118	3	651.2	**651.2**	227
C-50-r211	535.5	535.5	72	1	535.5	535.5	72
C-50-rc101	1066.7	945.6	5	3	954.9	**954.9**	17
C-50-rc102	854.6	794.5	42	3	833.4	**833.4**	80
C-50-rc103	716.0	665.4	32	3	714.1	**714.1**	77
C-50-rc104	∞	573.7	326	3	599.2	**599.2**	3054
C-50-rc105	913.6	860.2	16	3	894.7	**894.7**	43
C-50-rc106	768.0	734.2	60	3	767.8	**767.8**	111
C-50-rc107	712.8	662.0	139	3	672.4	**672.4**	214
C-50-rc108	∞	580.0	151	3	632.6	**632.6**	355
C-50-rc201	730.5	716.6	33	1	716.6	**716.6**	32
C-50-rc202	636.3	631.7	267	3	636.3	636.3	443
C-50-rc203	∞	572.4	722	7	579.1	**579.1**	2629
C-50-rc204	∞	449.2	565	509	464.3	**523.3**	108014
C-50-rc205	671.1	652.3	219	3	657.5	**657.5**	315
C-50-rc206	610.0	610.0	14	1	610.0	610.0	14
C-50-rc207	560.2	560.2	153	1	560.2	560.2	152
C-50-rc208	476.7	459.4	676	53	476.7	476.7	14230
C-100-c101	862.5	830.4	21	3	847.5	**847.5**	52
C-100-c102	862.5	829.9	33	3	847.5	**847.5**	83
C-100-c103	∞	828.9	59	3	846.3	**846.3**	153
C-100-c104	∞	824.2	72	3	842.2	**842.2**	240

Instance	BKS	RDB	Rt (s)	Nodes	BDB	BPB	t (s)
C-100-c105	857.2	829.9	21	3	847.5	**847.5**	62
C-100-c106	851.5	830.4	25	3	847.5	**847.5**	79
C-100-c107	857.2	829.9	20	3	847.5	**847.5**	79
C-100-c108	∞	829.9	30	3	847.5	**847.5**	85
C-100-c109	∞	828.8	32	3	847.1	**847.1**	74
C-100-c201	∞	627.0	181	1	627.0	**627.0**	181
C-100-c202	∞	626.0	628	3	626.8	**626.8**	16349
C-100-c203	∞	617.8	1321	1	617.8	**617.8**	1320
C-100-c204	∞	591.1	1874	3	617.2	**617.2**	3730
C-100-c205	∞	623.8	354	3	624.1	**624.1**	1272
C-100-c206	∞	622.6	509	3	623.7	**623.7**	3335
C-100-c207	∞	622.1	678	3	622.1	**623.5**	6739
C-100-c208	∞	616.2	426	1	616.2	**616.2**	426
C-100-r101	1679.4	1632.0	10	3	1634.8	**1634.8**	27
C-100-r102	1470.0	1460.4	14	1	1460.4	**1460.4**	21
C-100-r103	1185.7	1184.8	23	3	1185.0	**1185.0**	50
C-100-r104	∞	947.3	214	115	955.4	**955.4**	8530
C-100-r105	1361.4	1355.0	19	3	1355.3	**1355.3**	65
C-100-r106	1228.0	1227.8	41	15	1228.0	1228.0	211
C-100-r107	1047.1	1045.1	104	5	1047.1	1047.1	577
C-100-r108	∞	911.1	306	3	913.6	**913.6**	687
C-100-r109	1146.9	1144.3	55	3	1146.9	1146.9	225
C-100-r110	1068.0	1068.0	157	3	1068.0	1068.0	266
C-100-r111	∞	1045.1	136	5	1048.0	**1048.0**	590
C-100-r112	∞	945.1	446	7	948.6	**948.6**	3244
C-100-r201	1144.3	1143.2	78	5	1143.6	**1143.6**	313
C-100-r202	1029.8	1026.1	323	9	1029.8	1029.8	3064
C-100-r203	∞	868.0	1514	5	870.8	**870.8**	4330
C-100-r204	∞	720.2	1313	97	721.6	**747.1**	79596
C-100-r205	∞	944.8	519	7	949.8	**949.8**	3305
C-100-r206	∞	866.5	688	29	870.5	**995.5**	34344
C-100-r207	∞	783.0	853	85	787.6	**827.7**	90306
C-100-r208	∞	686.0	2244	25	687.5	∞	32906
C-100-r209	∞	847.6	768	161	850.6	**877.6**	75146
C-100-r210	∞	893.2	1019	149	895.9	**913.1**	83470
C-100-r211	∞	736.3	1184	203	738.1	**801.4**	108000
C-100-rc101	1685.3	1619.8	22	1	1619.8	**1619.8**	28
C-100-rc102	1471.9	1442.3	84	1	1442.3	**1442.3**	97
C-100-rc103	∞	1229.2	114	7	1235.9	**1235.9**	1085
C-100-rc104	∞	1097.6	252	3	1100.7	**1100.7**	570
C-100-rc105	1615.5	1513.7	57	1	1513.7	**1513.7**	68
C-100-rc106	∞	1366.8	171	3	1372.7	**1372.7**	312
C-100-rc107	1207.8	1205.4	95	1	1205.4	**1205.4**	117
C-100-rc108	1114.2	1108.5	242	3	1114.2	1114.2	541
C-100-rc201	1261.8	1261.5	97	3	1261.8	1261.8	147
C-100-rc202	1092.3	1092.3	282	1	1092.3	1092.3	314
C-100-rc203	∞	922.7	904	11	925.2	**1007.9**	9702
C-100-rc204	∞	770.7	9956	29	774.5	**880.1**	68778
C-100-rc205	1154.0	1152.4	228	3	1154.0	1154.0	419
C-100-rc206	∞	1044.3	298	11	1052.2	**1052.2**	1955
C-100-rc207	∞	955.2	1061	15	963.2	**963.2**	6421
C-100-rc208	∞	763.9	2848	77	767.0	**800.7**	83268

References

Z. Akca, R. T. Berger, and T. K. Ralphs. A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In John W. Chinneck, Bjarni Kristjansson, and Matthew J. Saltzman, editors, *Operations Research and Cyber-Infrastructure*, pages 309–330, Boston, MA, 2009. Springer. ISBN 978-0-387-88843-9.

Maria Albareda-Sambola, Elena Fernández, and Gilbert Laporte. The capacity and distance constrained plant location problem. *Computers & Operations Research*, 36(2):597–611, 2009.

J. Aráoz. *Polyhedral neopolarities*. PhD thesis, University of Waterloo, Department of Computer Science, 1974.

Julián Aráoz, Lisa Evans, Ralph E Gomory, and Ellis L Johnson. Cyclic group and knapsack facets. *Mathematical Programming*, 96(2):377–408, 2003.

Roberto Baldacci and Aristide Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380, 2009.

Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, Oct 2008. ISSN 1436-4646. doi: 10.1007/s10107-007-0178-5. URL `https://doi.org/10.1007/s10107-007-0178-5`.

Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011a. doi: 10.1287/opre.1110.0975.

Roberto Baldacci, Aristide Mingozzi, and Roberto Wolfler Calvo. An exact method for the capacitated location-routing problem. *Operations Research*, 59(5):1284–1296, 2011b. doi: 10.1287/opre.1110.0989.

José-Manuel Belenguer, Enrique Benavent, Christian Prins, Caroline Prodhon, and Roberto Wolfler Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931 – 941, 2011. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2010.09.019. URL `http://www.sciencedirect.com/science/article/pii/S0305054810002145`.

Imen Ben Mohamed, Walid Klibi, Ruslan Sadykov, Halil Şen, and François Vanderbeck. The two-echelon stochastic multi-period capacitated location-routing problem. HAL 02987266, Inria, 2020.

Rosemary T. Berger, Collette R. Coullard, and Mark S. Daskin. Location-routing problems with distance constraints. *Transportation Science*, 41(1):29–43, 2007. doi: 10.1287/trsc.1060.0156.

Maurizio Boccia, Antonio Sforza, Claudio Sterle, and Igor Vasilyev. A cut and branch approach for the capacitated p-median problem based on fenchel cutting planes. *Journal of mathematical modelling and algorithms*, 7(1):43–58, 2008.

E Andrew Boyd. Generating fenchel cutting planes for knapsack polyhedra. *SIAM Journal on Optimization*, 3(4):734–750, 1993.

E Andrew Boyd. Fenchel cutting planes for integer programs. *Operations Research*, 42(1):53–64, 1994.

Sunil Chopra, Sangho Shim, and Daniel E. Steffy. A few strong knapsack facets. In Boris Defourny and Tamás Terlaky, editors, *Modeling and Optimization: Theory and Applications*, pages 77–94, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23699-5.

Claudio Contardo and Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129 – 146, 2014.

Claudio Contardo, Vera Hemmelmayr, and Teodor Gabriel Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research*, 39(12):3185 – 3199, 2012.

Claudio Contardo, Jean-François Cordeau, and Bernard Gendron. A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optimization*, 10 (4):263 – 295, 2013. ISSN 1572-5286. doi: https://doi.org/10.1016/j.disopt.2013.07.005. URL `http://www.sciencedirect.com/science/article/pii/S1572528613000339`.

Claudio Contardo, Jean-François Cordeau, and Bernard Gendron. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS Journal on Computing*, 26(1):88–102, 2014. doi: 10.1287/ijoc.2013.0549. URL https://doi.org/10.1287/ijoc.2013.0549.

Luciano Costa, Claudio Contardo, and Guy Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4):946–985, 2019.

Said Dabia, Stefan Ropke, and Tom Van Woensel. Cover inequalities for a vehicle routing problem with time windows and shifts. *Transportation Science*, 53(5):1354–1371, 2019.

Mohammad M. Fazel-Zarandi and J. Christopher Beck. Using logic-based benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS Journal on Computing*, 24(3):387–398, 2012.

Ricardo Fukasawa, Humberto Longo, Jens Lysgaard, Marcus Poggi de Aragão, Marcelo Reis, Eduardo Uchoa, and Renato F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3): 491–511, May 2006. ISSN 1436-4646. doi: 10.1007/s10107-005-0644-x. URL https://doi.org/10.1007/s10107-005-0644-x.

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

G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85, Jun 1983.

Gilbert Laporte and Yves Nobert. An exact algorithm for minimizing routing and operating costs in depot location. *European Journal of Operational Research*, 6(2):224 – 226, 1981. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(81)90212-5. URL http://www.sciencedirect.com/science/article/pii/0377221781902125.

Gilbert Laporte, Yves Nobert, and D. Arpin. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, 6(9):291–310, Sep 1986. ISSN 1572-9338. doi: 10.1007/BF02023807. URL https://doi.org/10.1007/BF02023807.

Gilbert Laporte, Yves Nobert, and Serge Taillefer. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science*, 22(3):161–172, 1988. ISSN 00411655, 15265447. URL http://www.jstor.org/stable/25768316.

Jens Lysgaard, Adam N. Letchford, and Richard W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, Jun 2004.

D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. In *Proceedings of the 17th IPCO*, pages 393–403. Springer, 2014.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, 2017a.

Diego Pecin, Artur Pessoa, Marcus Poggi, Eduardo Uchoa, and Haroldo Santos. Limited memory rank-1 cuts for vehicle routing problems. *Operations Research Letters*, 45(3):206 – 209, 2017b.

Artur Pessoa, Eduardo Uchoa, and Marcus Poggi de Aragão. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks*, 54(4):167–177, 2009.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183:483–523, 2020.

Bjørn Petersen, David Pisinger, and Simon Spoorendonk. *Chvátal-Gomory Rank-1 Cuts Used in a Dantzig-Wolfe Decomposition of the Vehicle Routing Problem with Time Windows*, pages 397–419. Springer US, Boston, MA, 2008.

U Pferschy, H Kellerer, and D Pisinger. *Knapsack Problems*. Springer, 2004.

Marcus Poggi and Eduardo Uchoa. Integer program reformulation for robust branch-and-cut-and-price algorithms. In Laurence Wolsey, editor, *Mathematical Programming in Rio*, pages 56–61, 2003.

Marcus Poggi and Eduardo Uchoa. *New Exact Algorithms for the Capacitated Vehicle Routing Problem*, chapter 3, pages 59–86. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014. doi: 10.1137/1.9781611973594.ch3. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch3`.

Sattrawut Ponboon, Ali Gul Qureshi, and Eiichi Taniguchi. Branch-and-price algorithm for the location-routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 86:1 – 19, 2016.

Christian Prins, Caroline Prodhon, and Roberto Wolfler Calvo. Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. *4OR*, 4(3):221–238, Sep 2006. ISSN 1614-2411. doi: 10.1007/s10288-006-0001-9. URL `https://doi.org/10.1007/s10288-006-0001-9`.

Edward Rothberg. An evolutionary algorithm for polishing mixed integer programming solutions. *INFORMS Journal on Computing*, 19(4):534–541, 2007.

Ruslan Sadykov and François Vanderbeck. BaPCod — a generic Branch-And-Price Code. Technical report HAL-03340548, Inria Bordeaux — Sud-Ouest, September 2021.

Ruslan Sadykov, Eduardo Uchoa, and Artur Pessoa. A bucket graph–based labeling algorithm with application to vehicle routing. *Transportation Science*, 55(1):4–28, 2021.

Said Salhi and Graham K. Rand. The effect of ignoring routes when locating depots. *European Journal of Operational Research*, 39(2):150 – 156, 1989. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(89)90188-4. URL `http://www.sciencedirect.com/science/article/pii/0377221789901884`.

Michael Schneider and Michael Drexl. A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1):389–414, Dec 2017. ISSN 1572-9338. doi: 10.1007/s10479-017-2509-0. URL `https://doi.org/10.1007/s10479-017-2509-0`.

Michael Schneider and Maximilian Löffler. Large composite neighborhoods for the capacitated location-routing problem. *Transportation Science*, 53(1):301–318, 2019. doi: 10.1287/trsc.2017.0770. URL `https://doi.org/10.1287/trsc.2017.0770`.

Christian Tilk, Katharina Olkis, and Stefan Irnich. The last-mile vehicle routing problem with delivery options. *OR Spectrum*, 43(4):877–904, 2021.

Dilek Tuzun and Laura I. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116(1):87 – 99, 1999.

Laurence A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.