Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Nonrobust Strong Knapsack Cuts for Capacitated Location Routing and Related Problems

Pedro Henrique Liguori, A. Ridha Mahjoub, Guillaume Marques, Ruslan Sadykov, Eduardo Uchoa

informs
https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–19
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Contextual Areas

Nonrobust Strong Knapsack Cuts for Capacitated Location Routing and Related Problems

Pedro Henrique Liguori,[a] A. Ridha Mahjoub,[a,b] Guillaume Marques,[c,d] Ruslan Sadykov,[c] Eduardo Uchoa[c,e,*]

[a] Laboratoire d'Analyse et de Modélisation de Systèmes pour l'Aide à la Décision, Université Paris-Dauphine (Paris Sciences & Lettres), Paris 75016, France; [b] Department of Statistics and Operations Research, College of Science, Kuwait University, Kuwait; [c] Institut National de Recherche en Sciences et Technologies du Numérique, Inria Centre at the University of Bordeaux, Talence 33405, France; [d] Atoptima SAS, Bordeaux 33000, France; [e] Departamento deEngenharia de Produção, Universidade Federal Fluminense, Niteroi-RJ 24210-240, Brazil

*Corresponding author

Contact: phliguori@gmail.com, https://orcid.org/0009-0004-0426-7527 (PHL); ridha.mahjoub@lamsade.dauphine.fr, https://orcid.org/0000-0002-1079-1892 (ARM); guillaume.marques@inria.fr, https://orcid.org/0000-0002-5462-0876 (GM); ruslan.sadykov@inria.fr, https://orcid.org/0000-0002-4862-2226 (RS); eduardo_uchoa@id.uff.br, https://orcid.org/0000-0002-8687-2613 (EU)

Received: June 29, 2021
Revised: June 14, 2022
Accepted: March 16, 2023
Published Online in Articles in Advance: June 6, 2023

Area of Review: Transportation

https://doi.org/10.1287/opre.2023.2458

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 nonrobust 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.

Funding: This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior [PrInt UFF 88881], the Conselho Nacional de Desenvolvimento Científico e Tecnológico [Grant 313601/2018-6], and the Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro [Faperj E-26/202.887/2017].

Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2023.2458.

Keywords: location routing • cutting planes • column generation

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, that is, 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 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 $E_{IJ} = I \times J$. 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 (Poggi and Uchoa 2014, 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 *nonrobust*. Each separated cut adds an extra resource to the labeling algorithm used in the pricing, making it harder.

Nonrobust 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, nonrobust 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)), despite being potentially very strong, are not really practical: A few dozen such cuts already make 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 to substantially reduce their negative impact on the pricing. Despite those improvements, those limited-memory cuts are still nonrobust 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 nonrobust 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 nonrobust limited-memory rank-1 cuts. However, the main methodological advance is the *route load knapsack cuts* (RLKCs), a new family of nonrobust 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 nonrobust 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 zero or one 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 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 online 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 multidepots to be opened and subject to vehicle capacities. The computational results reported show that they were able to solve instances with eight 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 five 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 five 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 cannot 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 Mixed Integer Linear Programming 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 multidepot 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. (2023) 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 nonoverlapping 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, for example, 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 that 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 three-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 $\overline{U} = V \setminus U$ and denote by $\delta(U, W) \subseteq E$ the set of edges containing one endpoint in U and the other in W. We denote $\delta(U, \overline{U})$ 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 of one 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, that is, a route of the form i-j-i, variable $x^i_{(i,j)}$ takes a value of two. 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 one 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:

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

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

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

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

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

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

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

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

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

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

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 that 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 of two. 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) and (6)–(8) suffice.

3.2. Additional Cuts

Because 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*

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

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 $(\overline{y}, \overline{z})$ 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 $\overline{y}_i > 0$,

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

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

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

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

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

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 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 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

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

Let $\overline{y}$ be the current fractional solution of (F) restricted to y and define a vector of variables α having dimension $|I|$:

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

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

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

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

The potential difficulty with that FC separation is the enumeration of the set H. A point $\boldsymbol{h} \in H$ is said to be minimal if there is no other point $\boldsymbol{h}' \in H$ such that $\boldsymbol{h}' \leq \boldsymbol{h}$. It can be seen that nonminimal points lead to redundant constraints in (18). Therefore, 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.

A generalization of rounded capacity cuts is the so-called *depot capacity cuts* (DCCs) 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 DCC 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 DCCs*. 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

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

The validity of improved DCCs can be proved by considering the two possible values for y_{i_1}. When $y_{i_1} = 1$, Inequality (21) reduces to (20). Conversely, 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)

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

Proof. First consider the case where $y_{i_1} = y_{i_2} = 1$. The right-hand side of (22) simplifies to $2r(S,R) + (2r(S, R\setminus\{i_1\}) - 2r(S, R\setminus\{i_1, i_2\}))$. Because $2r(S, R\setminus\{i_1\}) - 2r(S, R\setminus\{i_1, i_2\}) \leq 0$, 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 $(\overline{x}, \overline{y})$ 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 $\overline{y}_i = 1$ 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 $\overline{y}_i > 0$ 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 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: $\tilde{S}(R) = \{j \in J : \sum_{i\in R}\sum_{e\in\delta(j)}\overline{x}_e^i = 2\}$. 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 $\sum_{i\in I\setminus R}\sum_{e\in\delta(S\cup\{j\})}\overline{x}_e^i$ 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, that is, $\cup_{i\in I}\Omega(i)$, 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, that is, $d(\omega) = \sum_{j\in J(\omega)} d_j$. 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, that is, $c(\omega) = \sum_{e\in E} b_e^\omega c_e$, where b_e^ω is the number of times edge e appears in ω (this coefficient can have values of zero, one, or two; 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:

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

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

Performing those variable substitutions, the following extended formulation is obtained:

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

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

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

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

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

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

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

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

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

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

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, which would be translated as

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

would be redundant in (EF), because the definition of route implies that $\sum_{e\in\delta(j)} b_e^\omega = 2a_j^\omega$ for every $j \in J$ and $\omega \in \Omega$. Inequalities (27) are still called GUBs, whereas (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) to have an integer solution. Because 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 no more than one inequality. For a nonnegative vector $\boldsymbol{\alpha} \in \mathbb{Z}_+^{|J|}$ of multipliers, the following rank-1 cut is valid:

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

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 $\boldsymbol{\alpha}$ are the same, the corresponding inequality is called a subset-row cut. Jepsen et al. (2008) introduced three-row and five-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 nonrobust 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 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), that is, 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 that 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, 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. Therefore, 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 $I' \subseteq I$, $1 \leq |I'| \leq 4$: $g_{I'}^{\text{ND}} = \sum_{i \in I'} y_i$.
- The number of vehicles starting in depot $i \in I$: $g_i^{\text{VN}} = (1/2)\sum_{j \in J} x_{ij}^i$.
- The total number of vehicles: $g^{\text{TVN}} = \sum_{i \in I} g_i^{\text{VN}}$.
- The number of customers served from depot $i \in I$: $g_i^{\text{CN}} = \sum_{j \in J} z_{ij}$.
- Assignment of customer $j \in J$ to depot $i \in I$: z_{ij}.
- Participation of edge $e \in E$ in the solution: $g_e^{\text{EDG}} = \sum_{i \in I} x_e^i$.

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 multiphase 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 5,000 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 Nonrobust Cuts

In this section, we present a new family of valid inequalities derived from knapsack-like Constraints (28). These inequalities are nonrobust 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*, 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

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

The next theorem characterizes the nontrivial facets (those that are not described by nonnegativity inequalities $t_q \ge 0$, $1 \le q \le W$ or by $\sum_{q=1}^{W} q\, t_q \le W$ itself) of this polytope.

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

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

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

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

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

Constraints (39) and (40) define, respectively, the *complementarity* and *superadditivity* properties that should be satisfied by any inequality defining a nontrivial facet $\mathcal{P}_{\text{MKP}}(W)$. The superadditivity together with (37) implies the *monotonicity* property (the coefficients in vector $\boldsymbol{\xi}$ are nondecreasing) because $\boldsymbol{\xi}_q = 0 + \boldsymbol{\xi}_q = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_q \le \boldsymbol{\xi}_{q+1}$.

The master polytope $\mathcal{P}_{\text{MKP}}(W)$ corresponds to an integer knapsack problem with the right-hand side W where all possible left-hand side coefficients, from one to W, do exist. General integer knapsack problems do not have all possible left-hand side coefficients. However, all nontrivial facets of a general integer knapsack polytope with right-hand side W are projections (obtained by eliminating the nonexisting coefficients) of some facet of $\mathcal{P}_{\text{MKP}}(W)$ (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 $2t_2 + 3t_3 + 4t_4 \le 5$, $\boldsymbol{t} \in \mathbb{Z}_+^3$. 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 $\xi_1 = 0$, $\xi_5 = 1$, $\xi_1 + \xi_4 = 1$, $\xi_2 + \xi_3 = 1$, $2\xi_1 + \le \xi_2$, $\xi_1 + \xi_2 \le \xi_3$, $\xi_1 + \xi_3 \le \xi_4$, $\xi_1 + \xi_4 \le \xi_5$, $2\xi_2 + \le \xi_4$, $\xi_2 + \xi_3 \le \xi_5$. 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 $z = 7/6$. Therefore, by Theorem 2, $0.5t_2 + 0.5t_3 + t_4 + t_5 \le 1$ defines a facet of $\mathcal{P}_{\text{MKP}}(5)$. By eliminating the nonexisting coefficients one obtains $0.5t_2 + 0.5t_3 + t_4 \le 1$, 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 nonzero 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_q^i as an integer variable indicating how many routes with a total load of *exactly* q units, where $1 \le q \le W_i$, leave depot $i \in I$. Those variables θ are simply an aggregation of variables λ and can be expressed as

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

Then, Inequalities (28) imply that:

$$\sum_{q=1}^{W_i} q\, \theta_q^i \le W_i\, y_i, \quad i \in I. \tag{42}$$

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

Theorem 3. *Inequality $\boldsymbol{\xi t} \le 1$ defines a nontrivial facet of $\mathcal{P}_{\text{MKP}}(W_i)$ if and only if inequality $\boldsymbol{\xi\theta}^i \le y_i$ defines a nontrivial facet of* $\text{conv}\{(\boldsymbol{\theta}^i, y_i) \in \mathbb{Z}_+^{W_i} \times \{0,1\} : \sum_{q=1}^{W_i} q\, \boldsymbol{\theta}_q^i \le W_i y_i\}$.

However, 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 $\boldsymbol{\xi} \in \mathbb{R}_+^{W_i}$ satisfying (37)–(38) and (40), the inequality

$$\sum_{q=1}^{W_i} \xi_q\, \theta_q^i \le y_i, \tag{43}$$

is known as a *route load knapsack cut* (RLKC).

Theorem 4. *RLKC* (43) *is valid for* (EF).

Proof. Let $(\bar{\boldsymbol{\theta}}^i, \bar{y}_i) \in \mathbb{Z}_+^{W_i} \times \{0,1\}$ be an integer solution of (EF) restricted to the variables θ and y related to depot i. If $\bar{y}_i = 0$, then $\bar{\theta}_q^i = 0$ for all $1 \le q \le W_i$, 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 one and W_i. By the repeated application of (37)–(38) and (40) and because $\boldsymbol{\theta}^i$ satisfies (42), we have that

$$\sum_{q=1}^{W_i} \xi_q \overline{\theta}_q^i \le \xi\left(\sum_{q=1}^{W_i} q\overline{\theta}_q^i\right) \le \xi(W_i) = 1 = \overline{y}_i.$$

Thus, Inequality (43) is also satisfied. □

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

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

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

which are RLKCs. The supperadditive property of the coefficients follows from the fact that $\lfloor r\rfloor + \lfloor r'\rfloor \le \lfloor r + r'\rfloor$ 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 $\Omega = \Omega(i)$, $\boldsymbol{\theta} = \boldsymbol{\theta}^i$, $y = y_i$, and $W = W_i$. Let $(\overline{\boldsymbol{\theta}}, \overline{y})$ denote the current fractional solution.

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

Second, 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 $z = \sum_{i=1}^{W} \overline{\theta}_q \xi_q$, 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 nontrivial facet $\boldsymbol{\xi t} \le 1$ of $\mathcal{P}_{\text{MKP}}(W)$ is called an $1/k$*-facet* if k is the smallest possible integer such that

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

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

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

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

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

$$a_m + a_{m'} \ge a_{\lceil m+m'\rceil} \qquad \forall m \le m' \text{ with } \lceil m + m'\rceil \le k. \tag{48}$$

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

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

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 $2 \le a_1 \le a_2 \le a_3 \le (W+1)/2$. Values a_4, a_5, and a_6 are then obtained from Equalities (47). According to Constraints (48), for each triple (a_1, a_2, a_3), we verify $2a_1 \ge a_2$, $a_1 + a_2 \ge a_3$, $a_1 + 2a_3 \ge W + 1$ (obtained from $a_1 + a_3 \ge a_4$ or from $a_3 + a_3 \ge a_6$), and $2a_2 + a_3 \ge W + 1$ (obtained from $a_2 + a_3 \ge a_5$ or from $a_2 + a_2 \ge a_4$).

As an example, consider a case where $Q = 70$, $W = 140$ and a fractional solution where the nonzeros are $\overline{\theta}_{38} = 1/14$, $\overline{\theta}_{53} = 1/2$, $\overline{\theta}_{65} = 16/14$, $\overline{\theta}_{70} = 1/2$, and $\overline{y} = 277/280$. The best rounding cut, obtained with multiplier $\beta = 1/53$, is $\sum_{q=53}^{105}(1/2)\theta_q + \sum_{q=106}^{140}\theta_q \le y$, 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 $\sum_{q=38}^{52}(1/6)\theta_q + \sum_{q=53}^{87}(1/2)\theta_q + \sum_{q=88}^{102}(5/6)\theta_q + \sum_{q=103}^{140}\theta_q \le y$, yielding a violation of around 0.094.

To separate 1/8-inequalities $\boldsymbol{\xi}^{8-(a_m)}$, 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 $2 \le a_1 \le a_2 \le a_3 \le a_4 \le (W+1)/2$. Values a_5, a_6, a_7, and 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 $2a_1 \ge a_2$, $a_1 + a_2 \ge a_3$, $a_1 + a_3 \ge a_4$, $2a_2 \ge a_4$, $a_1 + 2a_4 \ge W + 1$, $a_2 + a_3 + a_4 \ge W + 1$, and $3a_3 \ge W + 1$.

To separate 1/10-inequalities $\boldsymbol{\xi}^{10-(a_m)}$, 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 $2 \le a_1 \le a_2 \le a_3 \le a_4 \le a_5 \le (W+1)/2$. Values a_6, a_7, a_8, a_9, and 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 $2a_1 \ge a_2$, $a_1 + a_2 \ge a_3$,

$a_1 + a_3 \geq a_4$, $a_1 + a_4 \geq a_5$, $2a_2 \geq a_4$, $a_2 + a_3 \geq a_5$, $a_1 + 2a_5 \geq W + 1$, $a_2 + a_4 + a_5 \geq W + 1$, $a_3 + 2a_4 \geq W + 1$, and $2a_3 + a_5 \geq W + 1$.

For each sequence (a_m), which verifies (46)–(48), we generate vector $\boldsymbol{\xi}^{k-(a_m)}$ and check if the corresponding Inequality (43) is violated by $\overline{\boldsymbol{\theta}}$. 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 $\boldsymbol{\xi}^{k-(a_m)}$, 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 $|\Phi| \ll W$).
- An $1/k'$-inequality is also an $1/k$-inequality if k' is a divisor of k. Therefore, 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 nonrobust 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 mth such constraint having dual variable τ_m. Using the fact that $z_{ij} = 1/2\sum_{e\in\delta(j)} x_e^i$, $i \in I$, $j \in J$ (Equations (3)), we can eliminate the z variables and express the robust constraint m in format:

$$\sum_{i\in I}\sum_{e\in E}\eta_e^{im}x_e^i + \sum_{i\in I}\iota_i^m y_i \leq b_m.$$

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

$$\overline{c}_e^i = c_e - \sum_{m=1}^{M}\eta_e^{im}\tau_m.$$

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 $\overline{c}_a$ of every pair of opposite arcs (u, v) and $(v, u) \in A$, corresponding to $e = (u, v) \in E$, is set to $\overline{c}_e^i$. We have a single capacity resource. The resource consumption of arc $(j, j') \in A$ is equal to $(1/2)d_j + (1/2)d_{j'}$, 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 bidirectional 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), and 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 nondecreasing 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 nondecreasing 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, $\overline{c}(L)$ 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 $j(L') = j'$, $q(L') = q(L) + (1/2)d_{j(L)} + (1/2)d_{j(L')}$ and $\overline{c}(L') = \overline{c}(L) + \overline{c}_{a'}$. 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, that is, $(\omega(L'), \omega) \in \Omega$, 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 nondecreasing 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 $\overline{c}(L) \leq \overline{c}(L')$.*

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 $\overline{c}(L) \leq \overline{c}(L')$, 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 bidirectional 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 $\mathcal{L}$: $B_1(j, \mathcal{L}) = \min_{L \in \mathcal{L}} \{\bar{c}(L)\}$. Because function $g(q)$ is nondecreasing, 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 thus 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, $g(q(L) + q(L'))$, that is, 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. Therefore, 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) that 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; and
- IBM CPLEX Optimizer version 12.10 as the LP and MIP solver.

Experiments are run on a single thread of a 2× 18-core Cascade Lake Intel Xeon Skylake Gold 6240 server at 2.6 GHz having 196 GB of RAM.

The data sets and the source code needed to reproduce the experimental results presented in this paper are included as part of the online appendix. The data sets and the source code are also published in the Git repository at https://github.com/inria-UFF/location-routing.

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: RCCs 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. Comparison of Variants of the BCP Algorithm on CLRP Instances PPW06

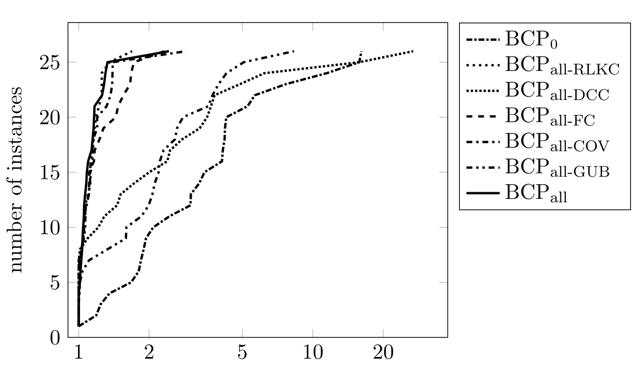
	Root					
Variant	Gap	Time (s)	Nodes	Average time (s)	Geomean time (s)	Solved
BCP_0	4.46%	57.9	19.2	6,373.8	758.7	24/26
$BCP_{all-GUB}$	3.08%	99.0	9.0	3,314.6	481.0	25/26
$BCP_{all-DCC}$	0.85%	101.0	9.1	5,995.1	504.6	24/26
BCP_{all-FC}	0.67%	111.4	4.4	3,174.7	283.9	25/26
$BCP_{all-RLKC}$	0.52%	114.7	4.1	2,994.8	264.0	25/26
$BCP_{all-COV}$	0.49%	114.4	4.6	3,172.6	273.4	25/26
BCP_{all}	0.48%	115.0	4.1	3,109.0	265.5	25/26

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.

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.

Figure 1. Performance Profiles of BCP Variants on CLRP Instances PPW06



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: $\rho = Q/(\sum_{i\in I} W_i/|I|)$. 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 that 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, that is, 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}, $BCP_{all-RLKC}$, and $BCP_{all-DCC}$ 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

Table 2. Comparison of Variants of the BCP Algorithm on Modified Instances PPW06 with Different Value ρ

		Root					
Variant	ρ	Gap	Time (s)	Nodes	Average time (s)	Geomeantime (s)	Solved
BCP_0	0.3	2.14%	32.1	18.6	4,290.3	344.7	19/20
$BCP_{all-DCC}$	0.3	0.74%	41.0	12.7	4,203.0	260.9	19/20
$BCP_{all-RLKC}$	0.3	0.51%	48.2	6.1	3,148.5	174.1	19/20
BCP_{all}	0.3	0.46%	49.3	5.8	3,057.3	162.9	19/20
BCP_0	0.5	3.33%	42.9	76.7	15,873.3	2,513.6	13/17
$BCP_{all-DCC}$	0.5	2.09%	108.4	35.0	12,963.9	1,979.7	13/17
$BCP_{all-RLKC}$	0.5	1.73%	69.8	24.3	11,155.1	1,059.3	14/17
BCP_{all}	0.5	1.26%	120.3	13.2	8,248.7	813.6	15/17
BCP_0	0.7	5.94%	51.6	255.3	28,272.8	10,511.0	6/17
$BCP_{all-DCC}$	0.7	2.49%	247.7	58.1	16,503.2	4,531.6	12/17
$BCP_{all-RLKC}$	0.7	3.91%	83.0	89.4	22,259.4	5,438.7	10/17
BCP_{all}	0.7	1.53%	284.6	18.9	9,491.4	1,734.7	14/17

the initial primal bound. For 51 of 54 instances, 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.

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.

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 used 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.

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

Table 3. Cut Generation Statistics for Original and Modified Instances PPW06

		Modified instances		
Cut family	Original instances	ρ = 0.3	ρ = 0.5	ρ = 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)	22,405.5 (153.0)	35,692.0 (223.8)	35,610.9 (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)	1,135.1 (10.1)	1,341.9 (14.8)	1,636.1 (11.4)
DCC1	419.1 (8.9)	1,078.0 (9.2)	1,320.3 (14.8)	1,622.4 (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)	11,726.6 (17.9)	12,785.2 (29.6)
RLKCround	26.8 (0.7)	296.1 (3.7)	1,304.8 (4.1)	109.9 (0.8)
RLKC1/2	0.7 (0.0)	0.5 (0.0)	757.2 (1.8)	1,401.8 (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)	1,115.7 (0.9)	1,738.4 (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)	2,318.9 (3.3)	2,611.1 (4.9)
RLKC1/10	13.8 (0.3)	16.6 (0.1)	4,691.6 (4.9)	4,986.7 (8.2)

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	Average time (s)	Gm. time (s)	Solved	Average time (s)	Gm. time (s)
PPW06	24/26	14,768	518	16/26	14,235	836
TB99	9/9	4,834	945	6/9	46,290	5,589

Note. Gm, Geometric mean time.

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, however, 24 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 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 Online Appendix EC.2.

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, that is, 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.

As 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 300-20-1 e, 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 Online Appendix EC.2.

Table 5. Performance of BCP_{best} on Instances in the Set SL19

Instances				
$\lvert I\rvert$	$\lvert J\rvert$	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%

6.2 VRP-CMD Instances

In this section, we test our algorithm on VRP-CMD instances that arise when solving the cut generation subproblem of the two-echelon stochastic multiperiod capacitated location-routing problem by a logic-based Benders decomposition approach (Ben Mohamed et al. 2023). 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 multidepot 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).

Table 6. Comparison of Variants of the BCP Algorithm on VRP-CMD Instances

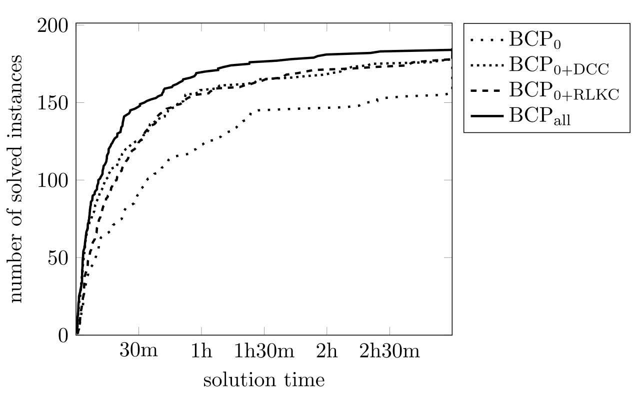
	Root					
Variant	Gap	Time (s)	Nodes	Average time (s)	Geomean time (s)	Solved
BCP_0	6.56%	12.0	194.0	3,422.4	1502.8	156/183
BCP_{0+RLKC}	6.41%	14.8	59.3	1,970.3	922.8	177/183
BCP_{0+DCC}	3.67%	22.2	58.0	1,847.9	672.4	177/183
BCP_{all}	3.57%	26.5	25.7	1,175.3	512.0	183/183

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 that 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 three 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 Online Appendix EC.3.

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.

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.

Figure 2. Performance Profile for BCP Variants Tested on 183 VRP-CMD Instances with Known Optimal Solutions



Online Appendix EC.3 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`, and `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. Four hundred seventy four 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 Online Appendix EC.4.

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 of 504 instances.

Now, as in the previous section, we test four BCP variants: BCP_0, BCP_{0+RLCK}, BCP_{0+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

Table 7. Comparison of Variants of the BCP Algorithm on VRPTW with Shifts Instances

		Root					
Variant	$\|J\|$	Gap	Time (s)	Nodes	Average time (s)	Geomean 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	2,158.5	304.8	123/139
BCP_{0+DCC}	100	0.84%	112.2	4.3	1,631.9	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

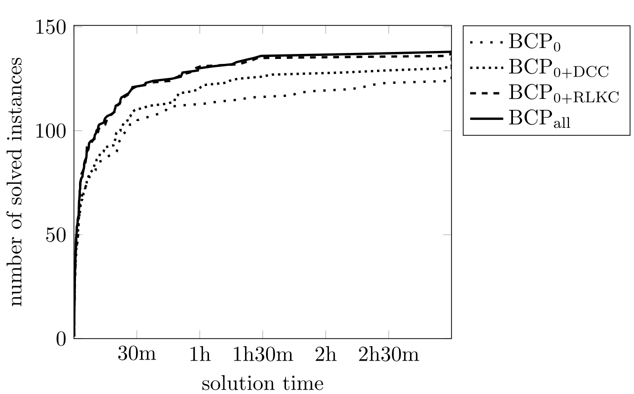
which we know the optimal solutions. The time limit is set to three hours.

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 also has a more modest, but still positive impact. This can be clearly seen in the performance profile in Figure 3.

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 nonrobust 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.

References

Akca Z, Berger RT, Ralphs TK (2009) A branch-and-price algorithm for combined location and routing problems under capacity restrictions. Chinneck JW, Kristjansson B, Saltzman MJ, eds. *Operations Research and Cyber-Infrastructure* (Springer, Boston), 309–330.

Albareda-Sambola M, Fernández E, Laporte G (2009) The capacity and distance constrained plant location problem. *Comput. Oper. Res.* 36(2):597–611.

Aráoz J (1974) Polyhedral neopolarities. PhD thesis, Department of Computer Science, University of Waterloo, Ontario, Canada.

Aráoz J, Evans L, Gomory RE, Johnson EL (2003) Cyclic group and knapsack facets. *Math. Programming* 96(2):377–408.

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

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

Baldacci R, Mingozzi A, Roberti R (2011a) New route relaxation and pricing strategies for the vehicle routing problem. *Oper. Res.* 59(5):1269–1283.

Baldacci R, Mingozzi A, Wolfler Calvo R (2011b) An exact method for the capacitated location-routing problem. *Oper. Res.* 59(5):1284–1296.

Belenguer J-M, Benavent E, Prins C, Prodhon C, Wolfler Calvo R (2011) A branch-and-cut method for the capacitated location-routing problem. *Comput. Oper. Res.* 38(6):931–941.

Ben Mohamed I, Klibi W, Sadykov R, Şen H, Vanderbeck F (2023) The two-echelon stochastic multi-period capacitated location-routing problem. *Eur. J. Oper. Res.* 306(2):645–667.

Berger RT, Coullard CR, Daskin MS (2007) Location-routing problems with distance constraints. *Transportation Sci.* 41(1):29–43.

Boccia M, Sforza A, Sterle C, Vasilyev I (2008) A cut and branch approach for the capacitated p-median problem based on fenchel cutting planes. *J. Math. Modeling Algorithms* 7(1):43–58.

Boyd EA (1993) Generating Fenchel cutting planes for knapsack polyhedra. *SIAM J. Optim.* 3(4):734–750.

Boyd EA (1994) Fenchel cutting planes for integer programs. *Oper. Res.* 42(1):53–64.

Chopra S, Shim S, Steffy DE (2015) A few strong knapsack facets. Defourny B, Terlaky T, eds. *Modeling and Optimization: Theory and Applications* (Springer International Publishing, Cham, Switzerland), 77–94.

Contardo C, Martinelli R (2014) A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optim.* 12:129–146.

Contardo C, Cordeau J-F, Gendron B (2013) A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optim.* 10(4):263–295.

Contardo C, Cordeau J-F, Gendron B (2014) An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS J. Comput.* 26(1):88–102.

Contardo C, Hemmelmayr V, Crainic TG (2012) Lower and upper bounds for the two-echelon capacitated location-routing problem. *Comput. Oper. Res.* 39(12):3185–3199.

Costa L, Contardo C, Desaulniers G (2019) Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Sci.* 53(4):946–985.

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

Fazel-Zarandi MM, Beck JC (2012) Using logic-based benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS J. Comput.* 24(3):387–398.

Fukasawa R, Longo H, Lysgaard J, de Aragão MP, Reis M, Uchoa E, Werneck RF (2006) Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. Programming* 106(3):491–511.

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

Laporte G, Nobert Y (1981) An exact algorithm for minimizing routing and operating costs in depot location. *Eur. J. Oper. Res.* 6(2): 224–226.

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

Laporte G, Nobert Y, Arpin D (1986) An exact algorithm for solving a capacitated location-routing problem. *Ann. Oper. Res.* 6(9):291–310.

Laporte G, Nobert Y, Taillefer S (1988) Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Sci.* 22(3):161–172.

Lysgaard J, Letchford AN, Eglese RW (2004) A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Programming* 100(2):423–445.

Pecin D, Pessoa A, Poggi M, Uchoa E (2014) Improved branch-cut-and-price for capacitated vehicle routing. *17th Conf. Integer Programming and Combinatorial Optimization* (Springer, Berlin), 393–403.

Pecin D, Pessoa A, Poggi M, Uchoa E (2017a) Improved branch-cut-and-price for capacitated vehicle routing. *Math. Programming Comput.* 9(1):61–100.

Pecin D, Pessoa A, Poggi M, Uchoa E, Santos H (2017b) Limited memory rank-1 cuts for vehicle routing problems. *Oper. Res. Lett.* 45(3): 206–209.

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

Pessoa A, Sadykov R, Uchoa E, Vanderbeck F (2020) A generic exact solver for vehicle routing and related problems. *Math. Programming* 183:483–523.

Petersen B, Pisinger D, Spoorendonk S (2008) *Chvátal-Gomory Rank-1 Cuts Used in a Dantzig-Wolfe Decomposition of the Vehicle Routing Problem with Time Windows* (Springer, Boston).

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

Poggi M, Uchoa E (2003) Integer program reformulation for robust branch-and-cut-and-price algorithms. Wolsey L, ed. *Mathematical Programming in Rio* (Universidade Federal do Rio de Janeiro, Rio de Janeiro), 56–61.

Poggi M, Uchoa E (2014) *New Exact Algorithms for the Capacitated Vehicle Routing Problem* (Society for Industrial and Applied Mathematics, Philadelphia).

Ponboon S, Qureshi AG, Taniguchi E (2016) Branch-and-price algorithm for the location-routing problem with time windows. *Transportation Res. Part E Logist. Transportation Rev.* 86:1–19.

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

Rothberg E (2007) An evolutionary algorithm for polishing mixed integer programming solutions. *INFORMS J. Comput.* 19(4):534–541.

Sadykov R, Vanderbeck F (2021) BaPCod—A generic branch-and-price code. Technical report, Inria Bordeaux—Sud-Ouest, Talence, France.

Sadykov R, Uchoa E, Pessoa A (2021) A bucket graph–based labeling algorithm with application to vehicle routing. *Transportation Sci.* 55(1):4–28.

Salhi S, Rand GK (1989) The effect of ignoring routes when locating depots. *Eur. J. Oper. Res.* 39(2):150–156.

Schneider M, Drexl M (2017) A survey of the standard location-routing problem. *Ann. Oper. Res.* 259(1):389–414.

Schneider M, Löffler M (2019) Large composite neighborhoods for the capacitated location-routing problem. *Transportation Sci.* 53(1):301–318.

Tilk C, Olkis K, Irnich S (2021) The last-mile vehicle routing problem with delivery options. *OR Spectrum* 43(4):877–904.

Tuzun D, Burke LI (1999) A two-phase tabu search approach to the location routing problem. *Eur. J. Oper. Res.* 116(1):87–99.

Wolsey LA (1998) *Integer Programming* (John Wiley & Sons, Hoboken, NJ).

Pedro Henrique Liguori works in BCG Gamma São Paulo, Brazil. His research interests are in polyhedral approaches for combinatorial optimization problems.

A. Ridha Mahjoub is a professor of operations research at Kuwait University and professor emeritus at Université Paris-Dauphine, France. His research areas include the theory and applications of combinatorial optimization and polyhedral theory. He currently serves as editor-in-chief of *RAIRO-Operations Research*.

Guillaume Marques is a co-founder of Atoptima, a startup that designs and develops cutting-edge cloud-based application solvers for businesses requiring complex resource assignment optimization. His research interests are in decomposition approaches for solving combinatorial optimization problems.

Ruslan Sadykov is a research fellow in the EDGE team in Inria Centre at the University of Bordeaux. His research interests include integer programming, branch-cut-and-price algorithm, vehicle routing and scheduling applications.

Eduardo Uchoa is a full professor at Universidade Federal Fluminense and holds an Inria International Chair (2022–2026). His research interests include column generation-based algorithms, vehicle routing, network design problems, and polyhedral combinatorics.