Networks

WILEY

RESEARCH ARTICLE OPEN ACCESS

Exact Methods for the Split Delivery Vehicle Routing Problem With Two-Dimensional Loading Constraints

Kamyla Maria Ferreira[1] | Pedro Munari[1] | Thiago Alves de Queiroz[2] | Claudia Archetti[3] | Reinaldo Morabito[1]

[1]Production Engineering Department, Federal University of São Carlos, São Carlos, São Paulo, Brazil | [2]Institute of Mathematics and Technology, Federal University of Catalão, Catalão, Goiás, Brazil | [3]Department of Economics and Management, University of Brescia, Brescia, Italy

Correspondence: Kamyla Maria Ferreira (kamylamaaria@gmail.com)

Received: 20 June 2024 | **Revised:** 28 May 2025 | **Accepted:** 3 June 2025

Funding: The authors are thankful to the two anonymous reviewers whose comments helped improve an earlier version of this paper. The authors would also like to thank the support of the São Paulo Research Foundation (FAPESP) (grant numbers 2022/05803-3, 2022/10303-0, and 2023/07988-3); the State of Goiás Research Foundation (FAPEG); and the National Council for Scientific and Technological Development (CNPq) (grant numbers 405369/2021-2, 405702/2021-3, 408722/2023-1, 314079/2023-8, 315555/2023-8, 304618/2023-3, 444679/2024-3, and 407005/2024-2).

Keywords: branch-and-cut algorithm | constraint programming | flow formulation | split delivery | two-dimensional packing | vehicle routing problem

ABSTRACT

The Split Delivery Vehicle Routing Problem with Two-dimensional Loading Constraints (2L-SDVRP) integrates vehicle routing, split delivery, and two-dimensional packing constraints. In the 2L-SDVRP, customers can be served by multiple vehicles, and their demands consist of different two-dimensional rectangular items that must be packed in the vehicles' bases. The problem involves determining the least-cost routes that satisfy all customer demands while ensuring the feasible packing of items in each vehicle. We present tailored branch-and-cut (BC) methods for solving the 2L-SDVRP. One of the methods is based on an effective, relaxed two-index vehicle flow formulation that is newly introduced in this paper. To evaluate the performance of the BC methods, computational experiments were conducted using both benchmark instances and new realistic instances inspired by cases from Brazilian logistics companies. The results indicate the superior performance of the method based on the two-index formulation, which obtained optimal solutions for 14 more instances than the other approach on the benchmark instances. This method also performed better on newly created instances, improving solutions by 5.6% on average.

1 | Introduction

The Vehicle Routing Problem (VRP) is a widely known combinatorial optimization problem in logistics and transportation, with numerous real-world applications, including product delivery, waste collection, public transportation, and supply chain management [1]. This paper focuses on the Split Delivery Vehicle Routing Problem with Two-dimensional Loading Constraints (2L-SDVRP), a variant that integrates vehicle routing, split deliveries, and packing decisions. Applications of the 2L-SDVRP relate to situations in which items are distinct, have different sizes, and cannot be stacked because they are tall or fragile. By efficiently solving this problem, companies may improve their decision-making process for transporting products like heavy machinery, furniture, and pallets [2, 3].

The 2L-SDVRP combines the Capacitated Vehicle Routing Problem with Two-dimensional Loading Constraints (2L-CVRP) with the Split Delivery Vehicle Routing Problem (SDVRP), which are challenging VRP variants widely studied in the literature [4, 5]. The problem involves determining vehicle routes to deliver two-dimensional rectangular items associated with customer demands. A route is feasible only if the items delivered by the route are orthogonally packed and entirely contained in the vehicle's rectangular base. Additionally, items cannot overlap, and their position in the base must respect the delivery order. The latter is called multidrop requirements, which is justified by the presence of heavy or fragile products that should have free passage during unloading operations [6]. A customer can be visited by one or more vehicles, and thus, the items requested by a customer can be delivered in multiple visits if this reduces the cost. The goal is to determine a feasible route plan that minimizes the total distance traveled.

Despite its practical relevance, the 2L-SDVRP has received little attention in the literature. A branch-and-cut (BC) method was proposed in [7], based on a three-index vehicle flow formulation for a variant of the 2L-SDVRP that considers a green objective function, referred to as the G2L-SDVRP. The authors also applied their method to instances of the 2L-CVRP and 2L-SDVRP, which are the only exact approaches proposed thus far for the latter. Results on benchmark instances indicate that the use of split deliveries in the 2L-SDVRP leads to a cost reduction of approximately 1% compared to the 2L-CVRP solutions, with a maximum reduction of 8.33%. Overall, considering a total of 60 instances, the 2L-SDVRP improved the solution in 26 instances, so in almost half of the cases. However, the BC method had difficulties in obtaining high-quality solutions within reasonable computation times for all considered variants. It could prove the optimality of only small instances, with up to 21 customers. In [3], the authors proposed an enhanced neighborhood search for a variant of the 2L-SDVRP in which items can be orthogonally rotated during the loading and relocated during the unloading operation at customers. The results reported by the authors show that savings can be achieved by allowing split deliveries, especially when the total area of items of each customer increases, that is, the area of items associated with each customer is close to the total area available in each vehicle.

Given the lack of effective exact approaches for the 2L-SDVRP, we seek to bridge this gap by presenting new BC methods. One of the methods is based on a relaxed two-index vehicle flow formulation that we introduce in this paper. This type of formulation has been proven highly effective for various other SDVRP variants [8–10]. The contributions of this paper are summarized as follows:

i) A new BC method based on a relaxed two-index vehicle flow formulation that does not initially consider all constraints of the 2L-SDVRP. Whenever a candidate incumbent solution is identified, a procedure is called to check its feasibility for the 2L-SDVRP. This method relies on cuts to eliminate infeasible solutions. We also embed valid inequalities generated using the CVRPSEP library to improve the lower bound provided by the linear programming (LP) relaxation [11]. Additionally, the method resorts to a tailored procedure to check packing feasibility using heuristics, lower bounds, and a constraint programming (CP) formulation.

ii) An improved version of the BC method of [7], by incorporating valid inequalities for routing decisions.

iii) Three new classes of instances based on real-world applications concerning the delivery of pallets, providing practical and relevant benchmarks for practitioners and researchers to evaluate new, future methods for the 2L-SDVRP and related variants.

We conducted extensive computational experiments using the benchmark and newly created instances. The new BC method, based on the relaxed two-index formulation, presented a superior overall performance, obtaining proven optimal solutions for 14 more instances than other approaches. The relaxed formulation promoted better feasible solutions and significant reductions in computation times.

The remainder of this paper is organized as follows. Section 2 presents the relevant literature on the 2L-SDVRP and its variants. Section 3 defines the 2L-SDVRP and presents the two-index and three-index vehicle flow formulations considered in this paper. Section 4 details the BC algorithms, and Section 5 presents the results of computational experiments using benchmark and the newly created instances. Finally, Section 6 shows our concluding remarks and ideas for future research.

2 | Literature Review

This section reviews the literature concerning the 2L-SDVRP and related variants, including the SDVRP and 2L-CVRP.

2.1 | Split Delivery Vehicle Routing Problems

The SDVRP was introduced in [12, 13] and has received considerable attention since then [14, 15]. In this problem, a customer's demand can be served by one or more vehicles (split deliveries) if this brings advantages regarding travel costs and/or the number of vehicles used. Exact methods for the SDVRP include a branch-and-bound algorithm based on constraint relaxation [16], a cutting-plane algorithm [8], a dynamic programming algorithm [17], an iterative two-stage method [18], an algorithm combining column- and cut-generation [19], BC methods [9, 10, 20], and branch-price-and-cut approaches [5, 21]. Concerning heuristic and hybrid methods, there have been several approaches proposed in the last years, including local search methods [22], column-generation-based approaches [23], an iterated local search [24], and hybrid heuristics [25, 26].

Most approaches have considered the traditional SDVRP, but contributions have also addressed variants motivated by real-world applications [26, 27]. Moreover, there are studies addressing the SDVRP with time windows [10, 28–30]; a heterogeneous fleet of vehicles [31, 32]; stochastic demands [33]; multiple commodities [34, 35]; and the emission of pollutant gases [32, 36].

2.2 | Vehicle Routing Problems With Two-Dimensional Loading Constraints

The 2L-CVRP was introduced in [2], motivated by practical constraints encountered in the distribution of furniture, mechanical components, and household appliances. In this variant, each customer's demand is given by a set of items that cannot be stacked in the vehicle's base. Thus, only the rectangular bases of these items are considered during the loading operations. We refer the interested reader to the comprehensive surveys [37] and [38] for a more detailed discussion on the 2L-CVRP.

Among the few exact methods proposed for the 2L-CVRP, we find BC methods [2, 7, 39–42], and one branch-price-and-cut algorithm [43]. Heuristic approaches for the 2L-CVRP are more common and include tabu search [44–46], neighborhood-based search [47], local search [48], and simulated annealing [49, 50].

Additional constraints have been considered in related variants of the 2L-CVRP, such as time windows [51], facility location requirements [52], pickup and delivery [53], heterogeneous fleet of vehicles [54, 55], backhauls [56], and stochastic travel times or demands [57, 58]. Realistic variants that appear in the food and construction industries were investigated in [54] and [51], respectively.

2.3 | Split Delivery and Two-/Three-Dimensional Loading Constraints

To the best of our knowledge, only two publications have simultaneously addressed split delivery and two-dimensional loading constraints thus far in the literature. In [7], the authors consider four VRP variants with two-dimensional items, namely the 2L-CVRP, the 2L-SDVRP, and extensions of both problems incorporating green requirements regarding CO_2 emissions (G2L-CVRP and G2L-SDVRP, respectively). The authors proposed a BC method with specific procedures to identify subtours and check the feasibility of packing the items delivered in each route. These procedures are based on several approaches, including heuristics, lower bounds, and a CP model.

Differently from [7], a heuristic approach is proposed in [3], which consists of a neighborhood search with three neighborhood operators based on insertion and swap. They extended these operators to include split deliveries. Also, they did not consider multidrop requirements; that is, they do not guarantee that the position of items in the vehicles respects the delivery order. Thus, it may be necessary to rehandle and/or rotate items by 90 degrees during unloading operations.

A few studies have addressed variants with split delivery and three-dimensional loading constraints, and thus far, there are only heuristic approaches, namely: a one-stage local search [59]; a data-driven three-layer search [60]; a tabu search [61]; a local search [62]; and a column-generation based algorithm [31].

Therefore, based on the presented literature, there is a lack of contributions regarding the VRP with loading constraints and split deliveries, especially regarding the proposal of effective exact methods. Our paper differs from the previous studies in terms of the solution approach. While [3] presented a heuristic, we are concerned with exact solutions. Moreover, we assume the multidrop requirements, which are practical limitations in several real-world situations. Compared with [7], we adapt their three-index vehicle flow formulation and enhance their BC approach by adding valid inequalities. Furthermore, we introduce a new BC method considering a lighter but relaxed formulation, which has been successfully used to solve other split delivery variants [9, 10]. Different from these previous studies, we resort to tailored cuts for the 2L-SDVRP and apply an enhanced procedure to solve the packing subproblems related to the items delivered by each route. All these components contribute to improving previous literature results, resulting in a new state-of-the-art method for the 2L-SDVRP.

3 | Problem Description and Mathematical Models

We define the 2L-SDVRP as follows: Let $N_c = \{1, \dots, n\}$ be the set of $n > 0$ customers and $G = (N, A)$ be a complete directed graph, with the node set $N = \{0\} \cup N_c$ and the arc set $A = \{(i, j) | i, j \in N, i \neq j\}$. Node 0 represents the depot for which all routes have to start and end. Set $K = \{1, \dots, K_{max}\}$ represents the fleet of $K_{max} > 0$ identical vehicles available at the depot to serve the customers' demand for rectangular items. Each vehicle has a maximum load capacity of Q and a two-dimensional rectangular base whose area is $A_T = W \times L$, with W as the width and L as the length. Each arc $(i, j) \in A$ has a non-negative cost c_{ij}, and we assume the corresponding cost matrix satisfies the triangle inequality.

Each customer's demand j is given by the set R_j of rectangular items. Each item $r \in R_j$ is characterized by a specific width w_{jr}, length l_{jr}, weight p_{jr}, and area $a_{jr} = w_{jr} \times l_{jr}$. Thus, the total weight of the items required by customer j is $P_j = \sum_{r \in R_j} p_{jr}$ and the total area is $A_j = \sum_{r \in R_j} a_{jr}$.

The 2L-SDVRP consists of finding delivery routes with minimum transportation costs to meet all customers' demands. These routes need to respect the following routing requirements: (*R1*) each vehicle starts and ends its route at the central depot; (*R2*) a vehicle can execute at most one route; (*R3*) a customer can be served by more than one vehicle, but a vehicle can visit a customer only once; (*R4*) the vehicle capacity in terms of weight and area cannot be exceeded; (*R5*) each route can serve one or more customers; (*R6*) the demand of each customer must be completely satisfied. In addition, the following loading requirements must be respected: (*L1*) items cannot overlap when packed in the vehicle's rectangular base; (*L2*) items assigned to a vehicle must be fully contained inside this vehicle's rectangular base; (*L3*) items must be packed with their sides parallel to the vehicle base sides (orthogonal packing); (*L4*) items have a fixed orientation and cannot be rotated, that is, items must be loaded according to a predetermined orientation, which is part of the input instance and not a decision variable (this orientation cannot be modified afterward); and (*L5*) items cannot be rearranged during the unloading operation at customers (multidrop requirement). This means the sequence of visits must be compatible with the loading plan, also called the last-in-first-out (LIFO) constraint. This constraint ensures that when a customer is visited, all their items must be accessible without obstruction from items belonging to other customers scheduled for later delivery. Hence, an item can only be unloaded if no other item blocks (even partially) the unloading door of the vehicle. Consequently, all the items of customer i must be in a position of the vehicle base for which their removal is not

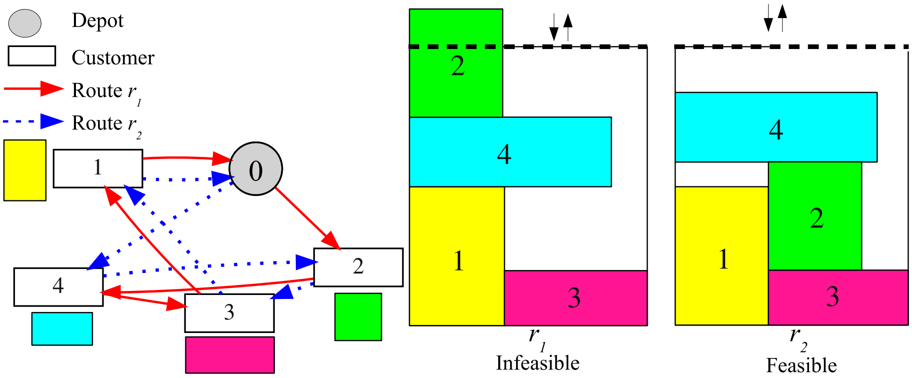

FIGURE 1 | Example of feasible and infeasible packing considering the LIFO constraints.

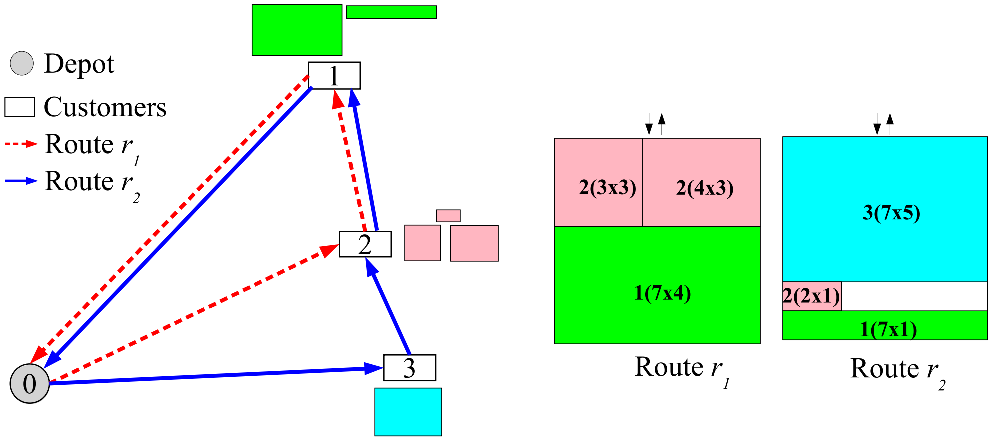

FIGURE 2 | Example of a unique optimal solution with two routes having two customers in common and arcs traversed more than once.

obstructed by any other item from a customer j, with $j \neq i$, who is served later in the route.

Figure 1 illustrates the packing associated with the LIFO constraint, highlighting the critical role of the customer service order in determining route feasibility. In the example shown in this figure, there are four customers, each requiring a single item. Two routes are presented for the same set of customers, but with different service sequences. Specifically, route r_1 is $0-2-4-3-1-0$ (solid line) and route r_2 is $0-4-2-3-1-0$ (dashed line). The packing associated with route r_1 is the leftmost depicted in Figure 1. It is infeasible because, according to the LIFO rule, the item for customer 4 must be loaded before the item for customer 2. Consequently, the item for customer 4 must not be obstructed by any item belonging to customers 2, 3, or 1, as these are scheduled for unloading at a later stage. However, this order makes it impossible to position item 2 in a feasible manner. Instead, when following the order of visit associated with route r_2, a feasible packing exists, as shown on the right in Figure 1.

Before presenting the formulations, it is worth focusing on an important difference between the 2L-SDVRP and the traditional SDVRP. In the literature, authors commonly rely on a few properties to strengthen the formulations of the SDVRP. According to [12, 13], in the SDVRP optimal solution, two routes have at most one customer in common, and any arc connecting two customers is traversed only once. However, these properties are not valid for the 2L-SDVRP. In Figure 2, we show an example of an optimal solution in which they are violated. Consider an instance with three customers and three vehicles. The vehicle base has dimension $(W, L) = (7, 7)$. Customer 1 demands two items, such that $(w_{11}, l_{11}) = (7, 4)$ and $(w_{12}, l_{12}) = (7, 1)$; customer 2 demands three items, where $(w_{21}, l_{21}) = (2, 1)$, $(w_{22}, l_{22}) = (3, 3)$ and $(w_{231}, l_{23}) = (4, 3)$; and customer 3 demands one item with $(w_{31}, l_{31}) = (7, 5)$. For the sake of simplicity, we consider a symmetric graph with the arc costs $c_{01} = c_{02} = c_{03} = c_{13} = 2$ and $c_{12} = c_{23} = 1$. The optimal solution in this example has the objective value 11, and two routes serve all customers. Customers 1 and 2 are served by two vehicles, and the arc (2, 1) is traversed twice. There is no alternative optimal solution for this example, so the properties mentioned above do not hold.

3.1 | Three-Index Vehicle Flow Formulation

We present a three-index vehicle flow (3IVF) formulation for the 2L-SDVRP, adapted from the model originally presented in [7] for the G2L-SDVRP. In addition to the notation introduced above, let Ω be the set of routes for which the assigned items and the loading plan are infeasible, that is, some of the conditions ($L1$)-($L5$) described above are not satisfied. Each element of this set is represented by a tuple $(\bar{S}, \bar{A}, \bar{R})$, associated with a route, in which $\bar{S}$ is the subset of customers served in the route; $\bar{A}$ is the subset of arcs traversed by the route; and $\overline{R}$ is the subset of items delivered to customers in the route. Additionally, K_{min} is a lower bound on the number of vehicles required to serve all customers in N_c (i.e., $K_{min} = \max\left\{\left\lceil \sum_{j\in N_c} P_j/Q \right\rceil, \left\lceil \sum_{j\in N_c} A_j/A_T \right\rceil\right\}$).

The 3IVF formulation considers the following binary variables: x_{ij}^k, which assumes the value of 1 if and only if arc $(i,j) \in A$ is traversed by vehicle $k \in K$; and z_{ir}^k, which assumes the value of 1 if and only if item $r \in R_i$ of customer $i \in N_c$ is served by vehicle $k \in K$. Using the presented notation, the 3IVF formulation is given as follows:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij} x_{ij}^k, \tag{1}$$

$$\text{s.t.} \sum_{j:(0,j)\in A} x_{0j}^k \leqslant 1, \qquad \forall k \in K : k > K_{min}, \tag{2}$$

$$\sum_{j:(0,j)\in A} x_{0j}^k = 1, \qquad \forall k \in K : k \leq K_{min}, \tag{3}$$

$$\sum_{i:(i,j)\in A} x_{ij}^k \leqslant 1, \qquad \forall j \in N_c;\ \forall k \in K, \tag{4}$$

$$\sum_{h:(h,i)\in A} x_{hi}^k = \sum_{j:(i,j)\in A} x_{ij}^k, \qquad \forall i \in N;\ \forall k \in K, \tag{5}$$

$$x_{ij}^k + x_{ji}^k \leqslant 1, \qquad \forall (i,j) \in A : i,j \in N_c;\ \forall k \in K, \tag{6}$$

$$\sum_{k\in K} z_{ir}^k = 1, \qquad \forall i \in N_c;\ \forall r \in R_i, \tag{7}$$

$$z_{ir}^k \leqslant \sum_{j\in N} x_{ji}^k, \qquad \forall i \in N_c;\ \forall r \in R_i;\ \forall k \in K, \tag{8}$$

$$\sum_{i\in N_c} \sum_{r\in R_i} p_{ir}\, z_{ir}^k \leqslant Q, \qquad \forall k \in K, \tag{9}$$

$$\sum_{i\in N_c} \sum_{r\in R_i} a_{ir}\, z_{ir}^k \leqslant A_T, \qquad \forall k \in K, \tag{10}$$

$$\sum_{i\in \overline{S}} \sum_{r\in R_i\cap \overline{R}} z_{ir}^k \leqslant |\overline{R}| - 1 + \sum_{(i,j)\in \bar{A}} (1 - x_{ij}^k), \quad \forall (\overline{S}, \bar{A}, \overline{R}) \subseteq \Omega; \forall k \in K, \tag{11}$$

$$\sum_{\substack{(i,j)\in A:\\ i,j\in \overline{S}}} x_{ij}^k \leqslant |\overline{S}| - 1, \qquad \forall \overline{S} \subseteq N_c : |\overline{S}| \geq 2;\ \forall k \in K, \tag{12}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall (i,j) \in A;\ \forall k \in K, \tag{13}$$

$$z_{ir}^k \in \{0,1\}, \qquad \forall i \in N_c;\ \forall r \in R_i;\ \forall k \in K. \tag{14}$$

The objective function (1) consists of minimizing the total travel cost. Constraints (2) ensure that the number of routes starting at the depot is limited to the total number of vehicles $|K| = K_{max}$, while constraints (3) impose at least K_{min} vehicles on being used. Constraints (4) impose that a vehicle can visit a customer at most once. Constraints (5) are classical flow conservation constraints. Constraints (6) impose that at most one of the arcs in two opposite directions, (i,j) or (j,i), $i,j \in N_c$, is traversed by the same vehicle k but not both, helping to eliminate subtours. Although constraints (12) imply (6), we decided to keep these constraints after conducting preliminary tests that demonstrated a positive impact. Constraints (7) ensure to satisfy all customer demands. Constraints (8) impose to assign an item requested by a customer only to a vehicle visiting that customer. Constraints (9) and (10) ensure that the vehicle's capacity in terms of weight and area is satisfied. Constraints (11) impose the infeasible-path inequalities, eliminating paths with an infeasible packing. Constraints (12) are subtour elimination constraints. Finally, the integrality conditions are given in constraints (13) and (14).

Due to the constraints of split delivery and packing, it is possible to visit a customer without delivering any item. To address this, in addition to constraints (2) to (14), the study in [7] includes additional constraints to ensure that at least one item is delivered whenever a customer is visited. However, rather than adopting this approach, we enforce the triangle inequality on costs, ensuring that each customer visit involves the delivery of at least one item. Both approaches were tested in this study, and preliminary computational experiments indicated that enforcing the triangle inequality is computationally less costly than introducing an additional constraint into the model.

To speed up solving the 3IVF formulation, we add the constraints (15) and (16), introduced in [7]. These constraints reduce the symmetry of the problem associated with a homogeneous fleet of vehicles while preserving an optimal solution. Hence, constraints (15) and (16) reduce the feasible solution space by removing redundancies arising from the problem's symmetry. Incorporating these constraints enables the method to more efficiently identify optimal or near-optimal solutions, as it avoids exploring solutions that are merely equivalent variations. In (15), the first vehicles serve the customers with the lowest indexes, that is, customer i can only be served by vehicle $k \in K \setminus \{1\}$ if at least one customer of index smaller than i is partially or completely served by vehicle $k-1$. Constraint (16) imposes that all or part of the demand of customer 1 is served by vehicle 1, that is, vehicle 1 always serves customer 1. Note that (16) is not needed and is inserted to strengthen the formulation.

$$\sum_{\substack{j\in N\\ j\neq i}} x_{ji}^k \leqslant \sum_{\substack{j\in N\\ j\neq i}} \sum_{\substack{h\in N_c\\ h\neq j, h\leq i}} x_{jh}^{k-1}, \quad \forall i \in N_c,\ \forall k \in K : k > 1, \tag{15}$$

$$\sum_{\substack{j\in N\\ j\neq i}} x_{j1}^1 = 1. \tag{16}$$

3.2 | Relaxed Two-Index Vehicle Flow Formulation

We now propose a relaxed two-index vehicle flow (R2IVF) formulation for the 2L-SDVRP. This formulation is relaxed in the sense that, as explained later, it might produce infeasible solutions. It makes use of the following decision variables: x_{ij}, is the number of times that an arc $(i,j) \in A$ is traversed by some vehicle; v_j, is the number of times that node $j \in N$ is visited. The formulation reads as follows:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}, \tag{17}$$

$$\text{s.t.} \sum_{i:(i,j)\in A}^{n} x_{ij} = v_j, \quad \forall j \in N, \tag{18}$$

$$\sum_{h:(h,i)\in A} x_{hi} = \sum_{j:(i,j)\in A} x_{ij}, \quad \forall i \in N, \tag{19}$$

$$\sum_{\substack{(i,j)\in A: \\ i\in \overline{S}, j\notin \overline{S}}} x_{ij} \geq \max\left\{ \left\lceil \sum_{i\in \overline{S}} \sum_{r\in R_i} \frac{p_{ir}}{Q} \right\rceil, \left\lceil \sum_{i\in \overline{S}} \sum_{r\in R_i} \frac{a_{ir}}{A_T} \right\rceil \right\}, \quad \forall \overline{S} \subseteq N_c, |\overline{S}| \geq 1, \tag{20}$$

$$x_{0j} \leq \min\{|R_j|, K_{max}\}, \quad \forall (0,j) \in A : j \in N_c, \tag{21}$$

$$x_{j0} \leq \min\{|R_j|, K_{max}\}, \quad \forall (j,0) \in A : j \in N_c, \tag{22}$$

$$x_{ij} \leq \min\{|R_i|, |R_j|\}, \quad \forall (i,j) \in A : i,j \in N_c, \tag{23}$$

$$1 \leq v_j \leq \min\{|R_j|, K_{max}\}, \quad \forall j \in N_c, \tag{24}$$

$$K_{min} \leq v_0 \leq K_{max}, \tag{25}$$

$$x_{ij} \in \mathbb{Z}_+, \quad \forall (i,j) \in A, \tag{26}$$

$$v_j \in \mathbb{Z}_+, \quad \forall j \in N. \tag{27}$$

The objective function (17) is related to minimizing the total travel cost. Constraints (18) link variables x_{ij} and v_j, ensuring that the number of visits to the node j is equal to the number of arcs that enter this node. Constraints (19) ensure that the number of arcs that enter a node equals the number of arcs that leave this node. Constraints (20) guarantee that the vehicle's capacity is respected, besides eliminating subtours. Constraints (21) to (24) impose an upper bound on the number of times that each arc can be traversed. These bounds consider the number of items a customer requests and the maximum number of available vehicles. Constraints (26) and (27) specify the domain of the decision variables.

Notably, the R2IVF formulation has no constraint related to the loading requirements (*L1*)-(*L5*) presented above for ensuring packing feasibility. Hence, a feasible solution obtained with this formulation may be infeasible for the 2L-SDVRP in terms of packing the items on each route. That is why we call it a *relaxed* formulation, as it includes solutions that are not feasible for the 2L-SDVRP. Hence, for an R2IVF solution to be considered feasible for the 2L-SDVRP, it is necessary to check the packing feasibility of that solution. Moreover, there is no variable in the R2IVF formulation that represents which items are transported by each route, which further complicates checking the feasibility of a solution when a customer is visited by two or more vehicles. Figure 3 presents an example illustrating the mentioned issues. There are five customers (numbered from 1 to 5) and two routes in this example, and the following arcs are traversed only once: (0, 1), (0, 4), (1, 2), (4, 2), (2, 3), (2, 5), (3, 0), and (5, 0). One vehicle departs from the depot and goes to customer 1, next to customer 2, and then has two possible destinations: customer 5 or 3. No information in this solution defines which of these two options is the next customer on this route, nor what items must be in each vehicle when they depart from customer 2. Hence, to check the feasibility of this solution regarding the 2L-SDVRP, we need to consider two different pairs of routes and verify the packing feasibility regarding different allocations of the items of customer 2.

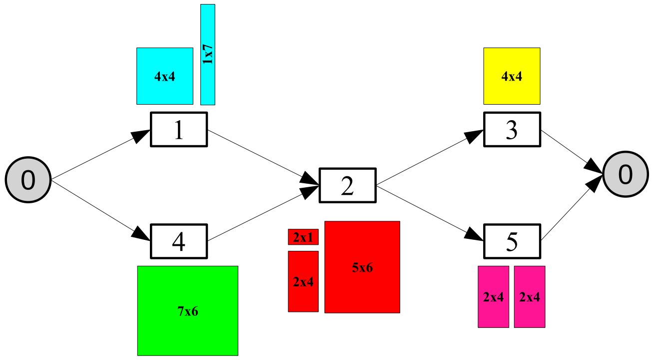

FIGURE 3 | Example of an R2IVF solution where a customer is visited twice without information about which customer items are transported on each route.

Figure 4 shows the two possible pairs of routes and their respective packing of items that may be derived from the R2IVF solution depicted in Figure 3. Figure 4a shows routes $0-4-2-3-0$ and $0-1-2-5-0$, while Figure 4b corresponds to routes $0-4-2-5-0$ and $0-1-2-3-0$. We consider that the vehicle base has dimension $(W, L) = (7, 10)$. Customer 1 demands two items, with $(w_{11}, l_{11}) = (4, 4)$ and $(w_{12}, l_{12}) = (1, 7)$; customer 2 demands three items where $(w_{21}, l_{21}) = (2, 1)$, $(w_{22}, l_{22}) = (2, 4)$ and $(w_{23}, l_{23}) = (5, 6)$; customer 3 demands one item, with $(w_{31}, l_{31}) = (4, 4)$; customer 4 demands one item where $(w_{41}, l_{41}) = (7, 6)$; and customer 5 demands two items where $(w_{51}, l_{51}) = (2, 4)$ and $(w_{52}, l_{52}) = (2, 4)$. The total area, summing over all the items, is 137, and since two vehicles are used, their capacities are merged. Thus, the total area available is 140, which is large enough to accommodate all items in terms of total area. However, given that a single item cannot be split and the packing of items in each vehicle must satisfy the loading requirements (L1)–(L5) presented above, the solution in Figure 3 is infeasible for the 2L-SDVRP, regardless of the combination of routes and items. Indeed, in both pairs of routes presented in Figure 4, the items from customers without split delivery are shown as already packed. Therefore, we need to decide how to split the items from customer 2 (in red) to obtain a feasible solution. However, it is impossible to pack the item (5×6) for customer 2 on either route. Consequently, the R2IVF solution shown in Figure 3 is infeasible for 2L-SDVRP.

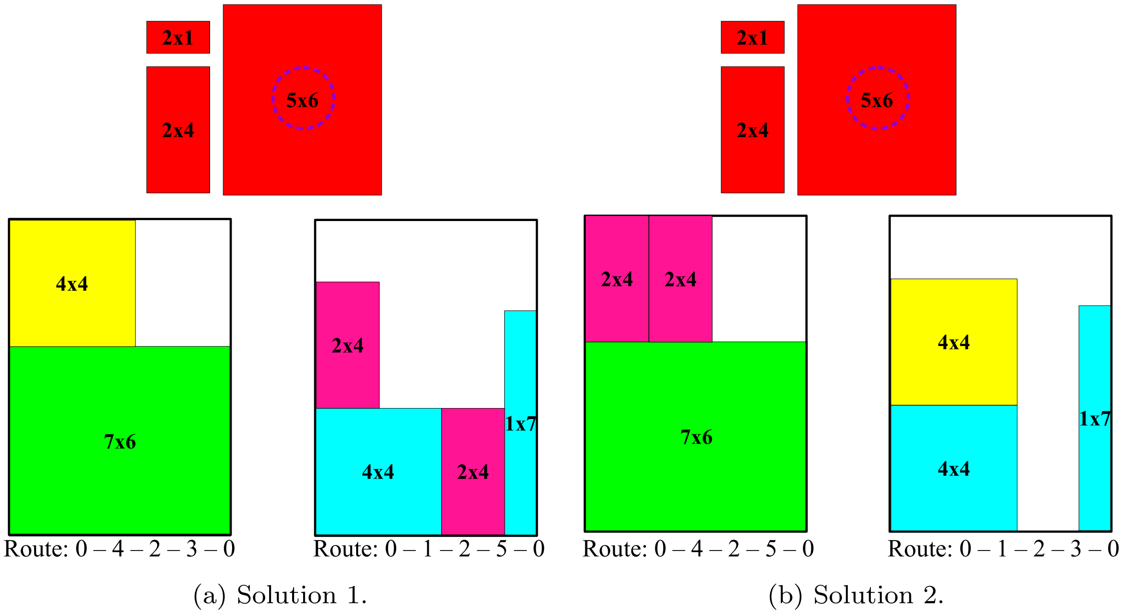

FIGURE 4 | Possible route solutions from the example in Figure 3. (a) Solution 1., (b) Solution 2.

To effectively identify appropriate solutions derived from this formulation for the 2L-SDVRP that adhere to the packing constraints, which have not been explicitly considered in R2IVF, we propose a BC method that uses cutting planes to eliminate infeasible solutions to the 2L-SDVRP from the feasible set of the R2IVF formulation. The subsequent section provides a detailed explanation of the methodology for assessing the feasibility of solutions obtained with the R2IVF formulation in the context of the 2L-SDVRP. Additionally, we discuss the procedures employed to evaluate whether a route has a feasible packing arrangement.

4 | Branch-And-Cut Algorithms

In this section, we describe BC algorithms for solving the formulations presented in Section 3. Specifically, Section 4.1 presents the algorithm for the 3IVF formulation, while Section 4.2 details the algorithm for the R2IVF formulation. We note that both formulations have an exponential number of constraints that are initially relaxed; thus, we use separation procedures to add dynamically when violated. Moreover, in Section 4.3, we describe the procedure used to solve the packing subproblem. As this procedure does not depend on the formulation, it is used in both BC algorithms.

4.1 | BC Algorithm for the 3IVF

The BC algorithm used to solve the 3IVF formulation is based on that introduced in [7] to solve the G2L-SDVRP. Different from what was done in the latter paper, we use the procedures available in the CVRPSEP library [11] to separate subtour elimination constraints (12). Our BC algorithm solves the 3IVF formulation without the packing feasibility constraints (11) and the subtour elimination constraints (12). During the solution process, we separate subtour elimination constraints for both fractional and integer solutions. We conducted tests to evaluate the impact of applying cuts either throughout the entire search procedure or exclusively at the root node. The results indicated that inserting cuts only at the root node leads to superior outcomes, achieving an average cost reduction of 0.21% for instances in which the exact approach was unable to prove optimality within the time limit, and an average decrease in computation time of 150 s.

For each candidate incumbent solution, after checking the subtours violation, the method invokes the separation procedure for detecting if each route in this solution has a feasible packing, as detailed in Section 4.3. Figure 5 illustrates the scheme of the overall separation procedure used by the BC algorithm for solving the 3IVF formulation. A more detailed explanation of the other steps in this algorithm can be found in [7].

4.2 | BC Algorithm for the R2IVF

The BC algorithm for the R2IVF formulation is new, and we now describe it in detail. For fractional and integer solutions, the method separates the capacity inequalities (20), considering area and weight separately, using the procedures available in the CVRPSEP library [11]. Additionally, to improve the performance of the BC method, if no violated capacity constraint is identified, it checks for the violation of the following connectivity constraints (28):

$$\sum_{i \in \overline{S}} \sum_{j \notin \overline{S}} x_{ij} \geq z_h, \quad \forall \overline{S} \subseteq N_c;\ |\overline{S}| \geq 1;\ \forall h \in \overline{S}. \tag{28}$$

These constraints are separated using the procedure described in [9].

As already mentioned, the R2IVF formulation in (18) to (27) cannot guarantee a feasible solution to the 2L-SDVRP, both in terms of packing (whose checking procedure is explained in Section 4.3) and in terms of the flow of items over nodes that are visited multiple times. Hence, whenever a candidate incumbent solution is

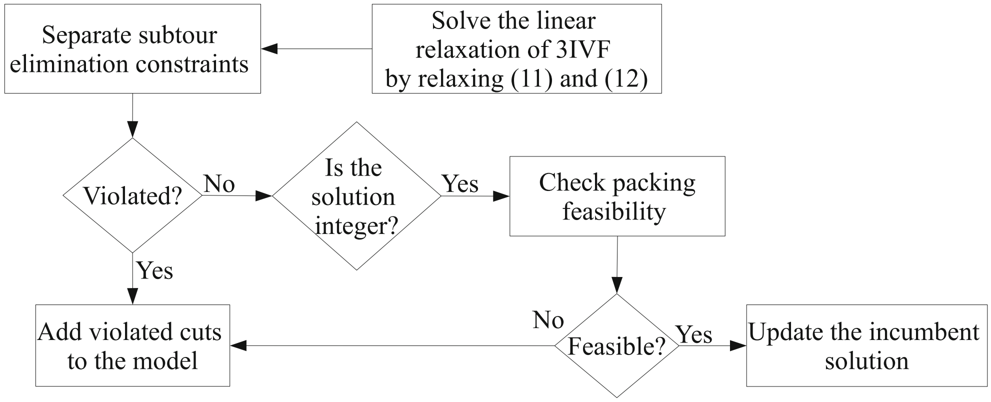

FIGURE 5 | Separation procedure of the BC algorithm for solving the 3IVF.

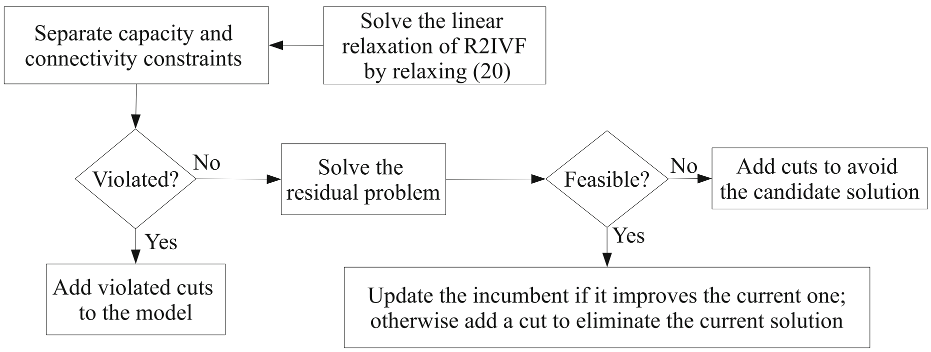

FIGURE 6 | Separation procedure of the BC algorithm for the R2IVF.

found, the BC method checks the flow and packing feasibility by solving a *residual problem*. This problem consists of a restricted version of the 3IVF formulation over fewer arcs. To solve this residual problem, we resort to the BC algorithm described in Section 4.1. However, note that the 3IVF formulation becomes much easier to solve in this setting, as many arcs are removed from the network. Figure 6 illustrates the separation procedure for each solution obtained by the BC method for the R2IVF formulation.

The residual problem is defined as follows: Let $\bar{x}$ be a candidate incumbent solution of the R2IVF formulation, with an objective value of $\bar{f}$. Let $\bar{z}$ be the value of the incumbent solution in the BC tree, that is, the best feasible solution found so far. We solve the 3IVF formulation where, for each pair of customers $i, j \in N_c$ such that $\bar{x}_{ij} = \bar{x}_{ji} = 0$, we fix $x^k_{ij} = x^k_{ji} = 0$ for $k = 1, \ldots, K_{max}$. For each pair of customers $i, j \in N_c$ where $\bar{x}_{ji} \geq 1$, or $\bar{x}_{ij} \geq 1$ and $\bar{x}_{ji} = 0$ (or vice versa), we preserve the original domains of these variables in the 3IVF formulation, that is, $x^k_{ij}, x^k_{ji} \in \{0, 1\}$ for $k = 1, \ldots, K_{max}$. Consequently, the key information transferred from R2IVF to 3IVF is whether arc (i, j) is traversed, determining which variables are fixed at zero in the 3IVF. Specifically, if arcs (i, j) and (j, i) are not traversed in the R2IVF solution, the variables associated with these arcs in the 3IVF formulation are fixed at zero, while the remaining variables continue to be unrestricted within their original domains. We also add a constraint to ensure that only solutions whose objective function is less than or equal to the incumbent objective value ($\bar{z}$) are considered.

When solving the residual 3IVF formulation, if a feasible solution with value $\bar{f}$ is found, we accept the candidate incumbent as the new incumbent in the R2IVF formulation and save the solution of the 3IVF formulation in an auxiliary structure that stores the best feasible solution for the 2L-SDVRP found so far. Otherwise, we add the valid cut (29) in the R2IVF formulation and continue with the BC method:

$$\sum_{i \in N_c} \sum_{\substack{j \in N_c: \\ \bar{x}_{ij} = 0 \\ \bar{x}_{ji} = 0}} x_{ij} \geq 1. \tag{29}$$

This cut ensures that at least one arc that is not part of the candidate incumbent solution needs to be traversed in a feasible solution of the 2L-SDVRP. Note that this cut is added in two situations: (a) when the residual problem is infeasible, meaning that there is no (improved) feasible solution of the 2L-SDVRP that has the same vehicle flow as in the candidate incumbent; and (b) when the optimal solution of the residual problem has a value different from $\bar{f}$, which means that this solution is not the same as the candidate incumbent (in terms of traversed arcs).

Cut (29) is a variant of the one proposed in [9]. In the latter paper, the authors only considered arcs between customers i and j that have not been traversed in the original direction (i, j). In contrast, we consider an arc between customers i and j only if it is not traversed in either the original direction (i, j) or the reverse direction (j, i). Therefore, this cut is stronger, as in [9], the summation in the left-hand side only considers $\bar{x}_{ij} = 0$, meaning the cut is given by $\sum_{i \in N_c} \sum_{\substack{j \in N_c \\ \bar{x}_{ij}=0}} x_{ij} \geq 1$. This way, a route traversed in the opposite direction could be a candidate solution, which is not the case for the cut (29) that we propose. Hence, we eliminate symmetric solutions when using cut (29), because we often encounter cases where both arcs (i, j) and (j, i) are traversed in the same solution. The example presented in Figure 2 can be used to illustrate this. In that example, there are two routes, namely $0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$ and $0 \rightarrow 2 \rightarrow 1 \rightarrow 0$. For simplicity, we assume in the example that the routes are not feasible for the residual problem and that the costs are symmetric. Hence, the cut proposed in [9] is $x_{12} + x_{13} + x_{23} + x_{31} \geq 1$. However, note that an alternative solution with routes $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$ and $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$ does not violate this cut and has the same value as the previous one, as we assumed costs are symmetric in this example. Conversely, both solutions are cut by (29), which corresponds to $x_{13} + x_{31} \geq 1$. It is important to note that this cut is valid for both symmetrical and asymmetrical costs.

4.3 | Checking the Packing Feasibility

This section presents the procedure for solving the packing subproblem regarding the items delivered in a given route, which is used in both BC algorithms presented above. We consider that the route is associated with S, where S is a subset of customers ($S \subseteq N_c$). For the sake of simplicity and with a slight abuse of notation, in the following, we will refer to route S, meaning the customers served on the route. The procedure comprises six methods based on calling constructive heuristics, calculating lower bounds, and solving Constraint Programming (CP) models. In addition to the set of customers served, each route S is also associated with the information concerning the set of items served per customer, defined as M_S, and the order in which the items in M_S are served on the route S, denoted as σ_S.

The packing subproblem associated with route S corresponds to the two-dimensional orthogonal packing problem with unloading constraints (2OPPUL). It considers a set of rectangular items, each defined by width, length, and unloading order. The unloading order specifies the sequence in which the items must be unloaded from the vehicle's rectangular base. Due to unloading constraints, items to be delivered first must be arranged for free passage during the unloading operations. This means that items of customers visited later on the route cannot block items of customers visited earlier and cannot be relocated during the unloading operations. In the 2OPPUL, a feasible solution consists of packing a set of rectangular items (with fixed orientation) into a two-dimensional orthogonal rectangular bin while satisfying unloading constraints, no overlap between any pair of items, and not extrapolating bin dimensions.

In the literature, different approaches have been proposed to determine the feasibility of a route according to the constraints mentioned above. Our approach relies on procedures that are similar to those presented in [7]; however, different from the latter paper, we use different heuristics (procedure 1 below), a lower bound based on dual feasible functions (procedures 2–4), and a procedure based on a grid of points, the so-called normal patterns [63], for the solution of the two-dimensional orthogonal packing problem (2OPP) (procedure 5) or 2OPPUL (procedure 6), considering the unloading requirements to determine the possible positions where items can be packed in the vehicle's base without modifying the unloading order. It is worth mentioning that other procedures were tested in preliminary experiments, and the best results, in terms of computation time, have been obtained using procedures 1 to 6. In addition, the order in which these procedures were applied was defined through preliminary tests, and we chose the one that led to the average shortest computation time. The following procedures are used to find a feasible solution or prove infeasibility:

Procedure 1: Apply three constructive heuristics from [49] in the following order: Bottom-Left Fill (W-axis), Bottom-Left Fill (L-axis), and Max Fitness Value. These heuristics pack items following a given sequence, which maintains the order of customer visits but might change the order of unloading the items per customer. For each heuristic, four rules are considered to generate sequences of items per customer by sorting them according to (*i*) nonincreasing order of their area; (*ii*) nonincreasing order of their width; (*iii*) nonincreasing order of their length; and (*iv*) nonincreasing order of their perimeter. Due to the multidrop requirement, rules (*i*)-(*iv*) are applied to each set of items with the same order of customer visit as defined in S. In other words, while the order of unloading the items of each customer may be modified using one of the rules (*i*)-(*iv*), the sequence of customer visits must remain unchanged. If any of these heuristics gives a feasible packing, the route has a feasible packing. Additionally, the procedure terminates when the first heuristic provides a feasible solution.

Procedure 2: Calculate the lower bounds based on the dual feasible functions described by [64]. They provide the minimum required length for packing all items. The route is infeasible if any lower bound is greater than L.

Procedure 3: Calculate the lower bound based on constraining the length positions described by [65]. The route is infeasible if this lower bound is greater than L.

Procedure 4: Calculate the lower bounds for the minimum width and length required to pack all items. They are based on solving the Gilmore-Gomory formulation of the cutting stock problem, as described by [65]. If the minimum width (or length) is greater than the width (or length) of the vehicle's rectangular base, then the route is infeasible.

Procedure 5: Solve the 2OPP, as described in [7]. It is a relaxed version of the 2OPPUL where the unloading constraints are not considered. The route is infeasible if no feasible packing is returned after solving this problem.

Procedure 6: Solve the 2OPPUL using CP formulation reported in [66]. This model refers to a non-preemptive cumulative-scheduling problem. To reduce the variables' domain, we use the grid of normal patterns introduced by [63], extending it to satisfy the unloading constraints.

ALGORITHM 1 | Solving the packing subproblem.

1 **Input:** S, M_S, σ_S, hash pool;
2 **Output:** Whether route S has a feasible packing;
3 $feas \leftarrow False$;
4 **if** *S is not in the hash pool* **then**
5 $P \leftarrow$ Procedures $\{1, 2, 3, 4, 5, 6\}$;
6 **foreach** $p \in P$ **do**
7 $feas \leftarrow$ Apply Procedure p;
8 **if** *feas is True* ***and*** *p is equal to Procedure 1* **then** Break the loop;
9 **else if** *feas is False* ***and*** *p is different of Procedure 1* **then** Break the loop;
10 Add S into the hash pool with the status in $feas$;
11 **else** $feas \leftarrow$ Status of the packing for the route S;
12 **return** $feas$;

The proposed procedure for checking the packing feasibility uses a hash structure to reduce the computational effort. This structure keeps the routes checked for the packing subproblem in memory. It follows the implementation described in [7], which has shown to be efficient.

Algorithm 1 describes the complete procedure used to check whether packing is feasible, considering the route S, the set of M_S items, and the order of delivery σ_S. The algorithm calls, for each route S, the procedures 1 to 6 sequentially until finding a feasible solution or proving infeasibility. For each route S, the initial step is to verify whether the route is already present in the hash structure. If it is, its status is returned (as indicated in line 11). If not, the strategies for verifying packing are applied (procedures 1 to 6). The first strategy employs heuristics (procedure 1); thus, if it is possible to pack all items while respecting the loading requirements (L1)–(L5), the packing is feasible. If packing is not feasible, the subsequent procedures (procedures 2 to 4) are employed to generate lower bounds for the dimensions of the vehicle base. If these lower bounds do not indicate infeasibility, meaning that the calculated lower bounds are not greater than the original dimensions of the vehicle base, the relaxation of the 2OPP model (procedure 5) is then solved. If the solution to the relaxed 2OPP model indicates that it is impossible to pack all items, the route is considered infeasible. This conclusion arises from the fact that some constraints of the original problem are relaxed; therefore, if packing is infeasible under the relaxed constraints, it can be inferred that it will also be infeasible in the original problem. Finally, if the relaxed 2OPP model fails to establish infeasibility, a constraint programming model (procedure 6) is employed to comprehensively assess the route's feasibility.

Note that the sequence in which procedures 1–6 are applied is such that procedure 6 is called only if the route is not proven feasible from procedure 1 or proven infeasible from procedures 2–5. This way, we reduce the number of calls to procedure 6, which is the most demanding in terms of computation time.

5 | Computational Results

The presented formulations and BC approaches were implemented in C++, using the BC libraries of the Gurobi Optimizer 10.0.1 and the CP optimizer libraries of the IBM ILOG CPLEX Optimization Studio 22.1. We performed experiments with each BC method with and without a warm-start initial feasible solution. We denote the BC methods without any warm-start as BC-3IVF and BC-R2IVF, according to their base formulation, whereas the methods that rely on warm-start are denoted as BC-3IVF* and BC-R2IVF*. For the latter, the same initial solution is used on both. Following [7], we obtain the initial solutions by solving the 2L-SDVRP without split deliveries (i.e., the 2L-CVRP), using exactly K_{max} vehicles and imposing that each route serves at least two customers. This formulation is simpler; thus, the solver can obtain a feasible solution in a relatively short computation time. Moreover, when solving this formulation, we set the solver priority to finding feasible solutions and impose a time limit of 300 s for each instance. The time spent in this process is subtracted from the total time limit of the BC method. Detailed information regarding the used initial solutions is provided in A, Tables A1 and A6.

All experiments reported in this section were performed on a Linux PC with an Intel Core i9-12900K 5.2 GHz processor and 128 GB of RAM. A time limit of 3600 s is set as the stop criterion for solving each instance. In both BC methods, each call to procedure 5 is limited to 60 s. We do not impose a time limit on the other procedures used to check packing because they run fast, except for procedure 6. Because procedure 6 is the last applied to solve the packing subproblem, it must be executed until the packing feasibility or infeasibility is definitively established.

In Section 5.1, we describe the instances used in our experiments, including both benchmark and newly generated instances. In Section 5.2, we compare the results of BC-3IVF* with those reported in [7] to evaluate the quality of the proposed method. Section 5.3 presents the results obtained on the benchmark instances, followed by a sensitivity analysis in Section 5.4. Finally, Section 5.5 discusses the results obtained on the new instances.

5.1 | Instance Classes

We perform computational experiments using two sets of instances. The first is the set of benchmark instances of the 2L-SDVRP [7], while the second is a new set of realistic instances we propose in this paper, based on real-world data from two Brazilian cases and considering different pallet standards, as detailed below. All instances are publicly available at https://bit.ly/taq. The instances in the first set are divided into six classes, namely classes 1 to 6. Classes 1 to 5 were proposed by [2], while class 6 was described in [7]. Instances in class 1 are not considered in our experiments because each customer requests only one item with dimensions (1, 1) in these instances. Thus, it is not possible to have split deliveries in their solutions. Each class contains 12 instances, giving a total of 60 instances.

Table 1 shows the items' characteristics for each class. It presents the range that encompasses the number of items demanded by each customer j ($|R_j|$) and the range of the items' dimensions concerning the length (l) and width (w), according to the vehicle's rectangular base measures. There are vertical items (i.e., the length is larger than the width), homogeneous (i.e., the length is close to the width), and horizontal (i.e., the length is smaller than

TABLE 1 | Items' characteristics in the 60 benchmark instances.

Class	$\|R_j\|$	Vertical l	Vertical w	Homogeneous l	Homogeneous w	Horizontal l	Horizontal w
2	$[1,2]$	$[0.4L, 0.9L]$	$[0.1W, 0.2W]$	$[0.2L, 0.5L]$	$[0.2W, 0.5W]$	$[0.1L, 0.2L]$	$[0.4W, 0.9W]$
3	$[1,3]$	$[0.3L, 0.8L]$	$[0.1W, 0.2W]$	$[0.2L, 0.4L]$	$[0.2W, 0.4W]$	$[0.1L, 0.2L]$	$[0.3W, 0.8W]$
4	$[1,4]$	$[0.2L, 0.7L]$	$[0.1W, 0.2W]$	$[0.1L, 0.4L]$	$[0.1W, 0.4W]$	$[0.1L, 0.2L]$	$[0.2W, 0.7W]$
5	$[1,5]$	$[0.1L, 0.6L]$	$[0.1W, 0.2W]$	$[0.1L, 0.3L]$	$[0.1W, 0.3W]$	$[0.1L, 0.2L]$	$[0.1W, 0.6W]$
6	$[2,4]$	$[0.1L, 0.6L]$	$[0.1W, 0.2W]$	$[0.1L, 0.3L]$	$[0.1W, 0.3W]$	$[0.1L, 0.2L]$	$[0.1W, 0.6W]$

TABLE 2 | Pallets' characteristics in the realistic instances.

Pallet type	$W(mm)$	$L(mm)$	$Q(kg)$
P_1	1200	1000	[300, 800]
P_2	1100	1100	[300, 800]
P_3	1200	800	[300, 800]
P_4	1200	1200	[300, 800]

TABLE 3 | Vehicles' characteristics in the realistic instances.

Vehicle type	$W(mm)$	$L(mm)$	$Q(kg)$
T_1	2480	14 370	14 000
T_2	2480	13 370	13 000
T_3	2480	12 470	12 000

the width). The vehicle's rectangular base measures 40 in length (L) and 20 in width (W). The distance between each pair of nodes is computed as the Euclidean distance rounded down to the nearest integer, and the Floyd–Warshall algorithm [67] is applied to guarantee that the triangle inequality holds. More details about these instances can be found in [2] and [7].

The instances in the second set were created based on the cases studied in [68], which involve real-world applications of the pallet loading problem. This problem concerns loading products onto pallets and then loading the pallets onto vehicles for distribution centers dealing with food and general products. Table 2 describes the different pallet types. For each type, the table shows the width and length of the pallet base and the range defining the pallet's minimum and maximum total weight when loaded. This range is used to randomly generate the actual weights of each customer's pallets. Pallet type P_1 is the Brazilian standard pallet; P_2 is the ISO Series 1 pallet recommended by the *International Standards Organization*; P_3 is the Europallet adopted by the *Union International des Chemins*; and P_4 is a bit larger square pallet than P_2 [68].

We consider three types of vehicles in the realistic instances, which are semitrailers commonly used in the Brazilian trucking industry. These types are presented in Table 3. For each vehicle type, the table shows the width and length of the vehicle's rectangular base and the maximum weight this vehicle can transport.

The created instances are grouped into three new classes, named classes 7 to 9, one for each vehicle type. Each class has 6 instances, one for each of the following number of customers: 10, 15, 20, 25, 30, and 35. For generating them, we assume that the central depot and customers are located in the Cartesian plane with the x-coordinates in the interval [0, 100], while the y-coordinates are in the interval [0, 200]. The number of pallets demanded per customer is in the range [2, 5], and the type of the pallet and its weight are randomly defined.

5.2 | Comparison With the Literature

We first compare the performance of BC-3IVF* with that of the original method proposed in [7] to assess the impact of the proposed changes. To ensure a fair comparison, the original method was executed on the same computer used for the experiments in the current study. Additionally, in contrast to the approach in the original paper, our implementation of their method ensures that the costs respect the triangle inequality, as this is also guaranteed in BC-3IVF*. Because a warm-start solution is provided in [7], we consider the BC-3IVF*, which also relies on warm-start. It is important to highlight that the initial solution provided for the method in [7] is identical to that of BC-3IVF*.

Table 4 presents the results for instances grouped into classes. For each instance class, the columns show the number of instances for which the solver finished with a proven optimal solution (OP); the number of instances for which the solver reached the time limit without proving optimality (TL), thus providing a feasible solution that can be nonoptimal; the average computation time, in seconds (T_{total}); the average computation time for checking the packing feasibility, in seconds (T_{pack}); the number of cuts including those for eliminating subtours (Cut_E) and those for eliminating routes with infeasible packing (Cut_P); and the average percentage optimality gap (Gap) reported by the solver, computed as $100 \times ((\text{Upper bound} - \text{Lower bound})/\text{Lower bound})$.

The results in Table 4 indicate that BC-3IVF* matches the method proposed in [7] in terms of optimal solutions, proving optimality for 20 instances. On average, BC-3IVF* reduced the computation time by 50 s compared to the literature approach, although the time required to solve the packing subproblem was higher. Regarding the number of cuts, BC-3IVF* introduced slightly more packing cuts than the literature method. However, the literature approach incorporated a significantly larger number of subtour elimination cuts than BC-3IVF*. Finally, on average, BC-3IVF* is closer to proving the optimality of instances that were stopped due to reaching the time limit compared to the literature approach.

TABLE 4 | Results obtained with BC-3IVF* and the approach proposed in [7].

	BC-3IVF*							Ferreira et al. [7]						
Class	OP	TL	$T_{total}(s)$	$T_{pack}(s)$	Cut_E	Cut_P	Gap	OP	TL	$T_{total}(s)$	$T_{pack}(s)$	Cut_E	Cut_P	Gap
2	3	9	2707.72	124.27	10 481	1364	15.92%	3	9	2713.01	6.49	34 074	1238	19.07%
3	3	9	2705.83	715.90	8817	699	14.50%	3	9	2707.41	17.58	29 014	433	16.75%
4	3	9	2705.62	668.45	7709	325	13.54%	4	8	2565.78	74.27	27 094	372	16.51%
5	5	7	2428.78	739.04	6144	109	11.49%	4	8	2462.97	417.12	22 838	72	15.70%
6	6	6	2097.04	597.11	5854	9	10.92%	6	6	2473.83	39.46	24 666	5	12.88%
2-6	20	40	2529.00	568.95	39 005	2506	13.27%	20	40	2584.60	110.98	137 686	2120	16.18%

TABLE 5 | Comparison of the solutions obtained with BC-3IVF* and with the approach proposed in [7].

	BC-3IVF* * Ferreira et al. [7]			
Class	B	E	W	Gap_{Sol}
2	1	11	0	0.00%
3	1	11	0	0.02%
4	4	6	2	0.15%
5	3	9	0	0.60%
6	1	11	0	0.37%
2-6	10	48	2	0.23%

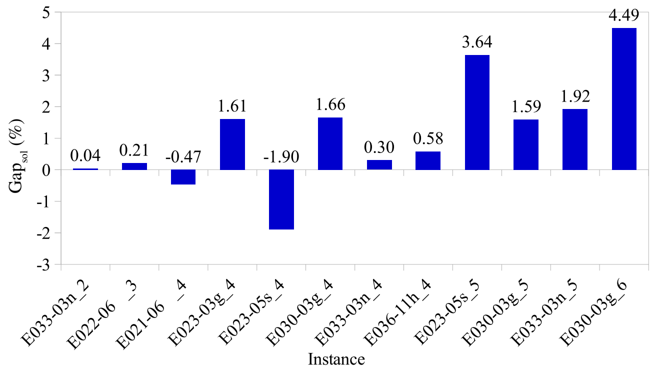

FIGURE 7 | Percentage gap for instances in which BC-3ICF* and the method in [7] have different solutions.

Table 5 presents, for each class of instances, the number of instances in which BC-3IVF* obtained better (B), equal (E), or worse (W) solutions than the literature method. Additionally, the table shows the average difference between the solutions (Gap_{Sol}), calculated as $100 \times (UB_{BC\text{-}3IVF^*} - UB_{Ferreira\ et\ al.\ [7]})/UB_{Ferreira\ et\ al.\ [7]}$, where $UB_{BC\text{-}3IVF^*}$ is the best solution value obtained with BC-3IVF* and $UB_{Ferreira\ et\ al.\ [7]}$ is the best solution value obtained with the literature method. A positive Gap_{Sol} indicates that the proposed approach outperforms the original. The results show that the solution values obtained with BC-3IVF* were, on average, 0.23% better than those obtained with the original method. Moreover, BC-3IVF* found 10 better solutions, 48 equal solutions, and 2 worse solutions.

Figure 7 illustrates the gap for instances where BC-3IVF* found solutions that differed from those obtained using the approach proposed in [7]. In the best case, the proposed method showed an improvement of 4.49% compared to the one from the literature. In the cases where its performance was inferior to the literature approach, the largest difference was around 1.90%, indicating a level of competitiveness comparable to the best available results.

5.3 | Comparison of BC Approaches on Benchmark Instances

This section reports the results of the BC approaches on the benchmark instances of the 2L-SDVRP (i.e., instances in classes 2 to 6). The results are summarized in Tables 6–8. Detailed results are presented in A. Table 6 brings the same information as Table 4. For BC-3IVF and BC-R2IVF, the table also gives the number of instances for which the corresponding BC method could not find any feasible solution within the time limit (WFS).

The results in Table 6 show that the BC methods based on the two-index formulation (i.e., BC-R2IVF* and BC-R2IVF) found more proven optimal solutions than those based on the formulation with three-index variables (i.e., BC-3IVF* and BC-3IVF). Specifically, BC-R2IVF* and BC-R2IVF proved optimality for 14 and 15 more instances than BC-3IVF* and BC-3IVF, respectively. The computation times for BC-R2IVF* and BC-R2IVF are shorter than for the other methods, on average, although they required longer times for solving the packing subproblems. These two methods also presented significantly smaller gaps than those based on the three-index formulation. Moreover, providing warm-start solutions for the BC methods was beneficial, as BC-3IVF and BC-R2IVF could not find feasible solutions for some instances within 3600 s. This happened in 9 instances for BC-3IVF and in 7 instances for BC-R2IVF. The gap values obtained with BC-R2IVF* are higher than those obtained with BC-R2IVF in classes 4 and 6. This discrepancy arises because, in these classes, BC-R2IVF failed to find a feasible solution within the time limit for two instances. Consequently, these instances, where no feasible solution was obtained, were excluded from calculating the average gap, thereby favoring the results of BC-R2IVF. Notably, the best overall results are achieved by BC-R2IVF*, while BC-3IVF shows the worst performance.

Table 7 reports the total number of cuts added in the separation procedures. For BC-3IVF* and BC-3IVF, these cuts include

TABLE 6 | Results on benchmark instances.

Class	BC-3IVF*					BC-3IVF					
	OP	TL	T_{total}(s)	T_{pack}(s)	Gap	OP	TL	WFS	T_{total}(s)	T_{pack}(s)	Gap
2	3	9	2708.35	3.08	15.92%	4	5	3	2682.30	10.10	20.30%
3	3	9	2706.78	16.01	14.50%	3	7	2	2711.33	33.56	28.22%
4	3	9	2705.81	34.69	13.54%	4	5	3	2699.46	76.88	10.27%
5	5	7	2429.28	485.23	11.49%	5	7	0	2516.45	541.43	21.58%
6	6	6	2099.32	27.51	10.92%	2	9	1	2548.75	565.29	15.81%
2-6	20	40	2529.91	113.30	13.27%	18	33	9	2631.66	245.45	19.24%
Class	**BC-R2IVF***					**BC-R2IVF**					
2	6	6	1940.34	21.77	4.24%	6	5	1	1987.37	22.13	8.04%
3	5	7	2191.42	129.21	3.55%	5	5	2	2225.09	175.66	5.70%
4	7	5	1660.41	419.86	2.43%	7	3	2	1738.53	454.42	1.74%
5	8	4	1261.35	575.39	1.89%	7	5	0	1314.01	698.14	2.32%
6	8	4	1262.52	414.73	0.57%	8	2	2	1238.63	385.93	0.21%
2-6	34	26	1663.20	312.19	2.53%	33	20	7	1700.72	347.26	3.60%

TABLE 7 | Number of cuts added on benchmark instances.

Class	BC-3IVF*		BC-3IVF		BC-2IVF*			BC-R2IVF		
	Cut_E	Cut_P	Cut_E	Cut_P	Cut_R	Cut_P	Cut_S	Cut_R	Cut_P	Cut_S
2	10 481	1364	27 667	4743	19 786	11 763	28 170	12 725	10 811	26 712
3	8817	699	29 401	1414	11 728	5966	34 454	8240	7182	24 087
4	7709	325	28 352	755	9923	2793	13 002	6276	3045	13 169
5	6144	109	23 440	133	8030	149	3842	7033	202	3318
6	5854	9	10 426	78	7473	106	1684	3953	67	2507
2-6	39 005	2506	119 286	7123	56 940	20 777	81 152	38 227	21 307	69 793

TABLE 8 | Comparison of BC-R2IVF* with the other methods on benchmark instances.

Class	BC-R2IVF				BC-3IVF*				BC-3IVF			
	B	E	W	Gap_{Sol}	B	E	W	Gap_{Sol}	B	E	W	Gap_{Sol}
2	3	8	1	3.32%	0	11	1	-0.23%	7	5	0	4.53%
3	7	5	0	1.54%	2	9	1	0.00%	8	4	0	9.27%
4	3	8	1	-0.08%	3	7	2	-0.03%	6	6	0	1.19%
5	3	8	1	0.49%	5	5	2	0.12%	5	5	2	5.30%
6	3	9	0	0.07%	3	9	0	0.41%	6	6	0	5.23%
2-6	19	38	3	1.07%	13	41	6	0.06%	32	26	2	5.10%

those for eliminating subtours (Cut_E) and those for eliminating routes with infeasible packing (Cut_P). Conversely, for BC-R2IVF and BC-R2IVF*, the cuts are designed to eliminate routes that violate capacity and connectivity constraints (Cut_R), as well as cuts related to the residual optimization problem (Cut_S). For BC-R2IVF* and BC-R2IVF, cuts to eliminate routes with infeasible packing are added when solving the residual optimization problem. After adding a cut, it is preserved for subsequent calls to the residual problem. The results presented in this table indicate that the most violated cuts relate to the residual optimization problem, especially for BC-R2IVF*. For BC-3IVF, more cuts related to subtour elimination are added compared to the route connectivity cuts in BC-R2IVF* and BC-R2IVF. At the same time, BC-R2IVF* and BC-R2IVF are associated with more cuts concerning routes with infeasible packing.

Table 8 compares BC-R2IVF* with the other BC methods. The columns show the number of instances in which the BC-R2IVF* solution is better (B), equal (E), or worse (W) than those obtained with BC-R2IVF, BC-3IVF*, and BC-3IVF, and the average percentage difference between the solutions (Gap_{Sol}). The Gap_{Sol}, for example, comparing BC-R2IVF* with BC-3IVF*, is given by $100 \times ((UB_{\text{BC-3IVF}^*} - UB_{\text{BC-R2IVF}^*})/UB_{\text{BC-R2IVF}^*})$, where $UB_{\text{BC-R2IVF}^*}$ and $UB_{\text{BC-3IVF}^*}$ are the upper bounds obtained by the BC-R2IVF* and BC-3IVF*, respectively. Notably, BC-R2IVF* obtained better solutions for 19, 13, and 32 instances when compared with BC-R2IVF, BC-3IVF*, and BC-3IVF, respectively. For more than half of the instances, the solutions obtained by BC-R2IVF* are better than or equal to those obtained by BC-R2IVF, BC-3IVF*, and BC-3IVF. For BC-3IVF*, we note solutions whose gaps are worse than those obtained with BC-R2IVF* but better than those obtained with BC-R2IVF or BC-3IVF.

5.4 | Sensitivity Analyses on Benchmark Instances

Tables 9–12 present a sensitivity analysis examining the impact of specific problem features on the results of R2IVF*. Table 9 reports the percentage of customers requiring more than one item (NI) for each group of instances. For each type of solution, which can be optimal (OP) or terminated due to the time limit (TL), where the latter includes cases in which the BC method found at least one feasible solution within the time limit but was unable to prove its optimality, the table shows the number of instances (NB_{ins}), the number of customers served with split deliveries (SD_{C}), and the number of instances where at least one split delivery occurred (SD_{ins}). The results indicate that, in both types of solutions, instances with more items per customer (classes 5 and 6) exhibit a higher frequency of split deliveries, as expected. Furthermore, almost all instances solved to optimality involved at least one customer being served with a split delivery. Specifically, in classes 5 and 6, approximately half of the instances included at least one split delivery. In contrast, in class 3, the method encountered greater difficulty in achieving optimal solutions and showed the fewest instances with split deliveries. While an increase in the number of items expands the range of potential solution configurations, split deliveries enable a more efficient arrangement of loads on packing, thereby facilitating the resolution of problems involving a high number of items.

Table 10 shows the influence of the items' dimensions on the results. Instances are categorized into three groups based on the predominant characteristics of the items requested by the majority of the customers: vertical (i.e., the length is greater than the width), homogeneous (i.e., the length is approximately equal to the width), or horizontal (i.e., the length is smaller than the width). The results indicate that instances with vertical items are the easiest to solve, followed by those with homogeneous items and, finally, those with horizontal items. When most items are in the category horizontal, their length is typically smaller than their width. This feature increases the difficulty of solving the packing subproblem due to the multidrop constraint, as it requires accommodating the items in a manner that adheres to the delivery sequence. Most of the instances with the highest number of vertical items that BC-R2IVF* could not solve within the time limit are those with 30 and 35 customers, which are the largest instances in the set.

Table 11 illustrates the impact of items' areas on problem resolution, where the area of an item is calculated as the product of its length by its width. For each class of instances, the table reports the number of instances solved to proven optimality (OP) and for

TABLE 9 | Numbers of split deliveries.

Instance		OP			TL		
Class	NI	NB_{ins}	SD_{C}	SD_{ins}	NB_{ins}	SD_{C}	SD_{ins}
2	58.57%	6	4	4	6	1	1
3	68.52%	5	6	3	7	5	2
4	66.92%	7	8	5	5	3	2
5	79.22%	8	12	6	4	2	1
6	100.00%	8	12	7	4	8	3
2-6		34	42	25	26	19	9

TABLE 10 | Influence of the items' dimensions.

Instance		OP				TL			
Class	NI	NB_{ins}	Vertical	Homogeneous	Horizontal	NB_{ins}	Vertical	Homogeneous	Horizontal
2	58.57%	6	6	0	0	6	1	4	1
3	68.52%	5	2	2	1	7	3	2	2
4	66.92%	7	4	3	0	5	3	2	0
5	79.22%	8	8	0	0	4	4	0	0
6	100.00%	8	8	0	0	4	4	0	0
2-6		34	28	5	1	26	15	8	3

TABLE 11 | Influence of the items' area.

	Status		OP						TL					
Class	**OP**	**TL**	**NI**	T_{total}	T_{pack}	**[0, 10]**	**[10, 20]**	**[20, 30]**	**NI**	T_{total}	T_{pack}	**[0, 10]**	**[10, 20]**	**[20, 30]**
2	6	6	58.57%	280.67	2.22	64.99%	32.71%	2.30%	44.68%	tl	41.33	50.70%	48.13%	1.17%
3	5	7	68.52%	219.40	6.91	76.13%	23.87%	0.00%	64.09%	tl	216.56	82.31%	17.69%	0.00%
4	7	5	66.92%	274.98	135.85	86.53%	13.47%	0.00%	73.70%	tl	817.47	85.60%	14.40%	0.00%
5	8	4	79.22%	92.02	24.21	98.29%	1.71%	0.00%	72.93%	tl	1677.74	99.05%	0.95%	0.00%
6	8	4	100.00%	93.78	86.28	98.02%	1.98%	0.00%	100.00%	tl	1071.62	97.10%	2.90%	0.00%
2-6	34	26	74.65%	192.17	51.09	84.79%	14.75%	0.46%	71.08%	tl	764.95	82.95%	16.81%	0.23%

TABLE 12 | Impact of the number of customers and items.

Instance		OP						TL						
N_c	**MI**	**NI**	NB_{ins}	SD_C	SD_{ins}	T_{total}	T_{pack}	**NI**	NB_{ins}	SD_C	SD_{ins}	T_{total}	T_{pack}	**Gap**
15	[20, 30)	63.34%	2	2	2	149.64	2.60	—	—	—	—	—	—	—
15	[30, 40)	73.33%	3	2	1	328.15	18.89	—	—	—	—	—	—	—
15	[40, 50)	88.00%	5	6	5	3.65	2.83	—	—	—	—	—	—	—
20	[20, 30)	—	—	—	—	—	—	45.00%	1	0	0	tl	36.31	10.96%
20	[30, 40)	60.00%	1	0	0	551.97	1.87	—	—	—	—	—	—	—
20	[40, 50)	65.00%	3	7	3	17.10	3.64	85.00%	1	0	0	tl	92.10	6.91%
20	[50, 60)	75.00%	1	2	1	92.22	1.10	—	—	—	—	—	—	—
20	[60, 70)	91.67%	3	9	3	268.27	201.04	—	—	—	—	—	—	—
21	[30, 40)	52.38%	3	2	2	11.32	2.12	—	—	—	—	—	—	—
21	[40, 50)	54.76%	2	1	1	82.14	3.36	—	—	—	—	—	—	—
21	[50, 60)	77.78%	3	2	2	1.19	0.28	—	—	—	—	—	—	—
21	[60, 70)	100.00%	2	2	1	2.33	0.28	—	—	—	—	—	—	—
22	[20, 30)	—	—	—	—	—	—	31.82%	1	1	1	tl	71.83	7.21%
22	[30, 40)	—	—	—	—	—	—	45.45%	1	0	0	tl	51.78	8.40%
22	[40, 50)	—	—	—	—	—	—	57.58%	3	2	2	tl	807.87	3.15%
22	[50, 60)	68.18%	1	1	1	1726.46	909.66	65.91%	2	0	0	tl	2449.11	6.60%
22	[70, 80)	100.00%	2	2	2	58.72	43.28	—	—	—	—	—	—	—
25	[40, 50)	60.00%	1	0	0	801.47	0.23	—	—	—	—	—	—	—
25	[60, 70)	—	—	—	—	—	—	78.00%	2	2	1	tl	0.71	4.55%
25	[70, 80)	—	—	—	—	—	—	100.00%	1	3	1	tl	0.41	1.37%
25	[90, 100)	92.00%	1	4	1	62.15	0.41	—	—	—	—	—	—	—
29	[40, 50)	—	—	—	—	—	—	46.56%	2	0	0	tl	287.73	9.27%
29	[70, 80)	—	—	—	—	—	—	79.31%	1	0	0	tl	839.45	8.43%
29	[80, 90)	—	—	—	—	—	—	82.76%	1	0	0	tl	1812.67	6.57%
29	[90, 100)	—	—	—	—	—	—	100.00%	1	2	1	tl	1340.97	0.49%
32	[40, 50)	—	—	—	—	—	—	37.50%	1	0	0	tl	51.74	12.67%
32	[50, 60)	—	—	—	—	—	—	53.13%	1	0	0	tl	284.16	10.61%
32	[70, 80)	—	—	—	—	—	—	71.88%	1	0	0	tl	1414.43	7.37%
32	[90, 100)	—	—	—	—	—	—	100.00%	1	3	1	tl	2945.05	0.98%
32	[100, 110)	84.38%	1	0	0	476.01	178.35	—	—	—	—	—	—	—
35	[60, 70)	—	—	—	—	—	—	60.00%	1	0	0	tl	0.04	3.96%
35	[70, 80)	—	—	—	—	—	—	77.14%	1	4	1	tl	0.26	3.50%
35	[90, 100)	—	—	—	—	—	—	77.14%	1	0	0	tl	8.71	5.50%
35	[100, 110)	—	—	—	—	—	—	100.00%	1	0	0	tl	0.04	3.96%
35	[110, 120)	—	—	—	—	—	—	77.14%	1	2	1	tl	0.09	2.91%

which the method reached the time limit without proving optimality, but found at least one feasible solution (TL). Additionally, for each type of solution, it provides the percentages of customers requiring more than one item; the total time spent by the method; the time dedicated to packing; and, for each item area range, the percentage of items with an area within that range. The range is measured as a percentage of the rectangular base of the vehicle.

The results indicate that instances with smaller items and/or a higher number of items with similar dimensions are easier to solve. The highest number of optimal solutions was achieved for classes 5 and 6, where more than 90% of the items occupy less than 10% of the vehicle's total area. The items of class 2 are the most diverse, with item areas reaching up to 30% of the vehicle's capacity. For these instances, the method proved optimality for only half of the instances. The worst performance of the method was observed for class 3, despite the absence of items occupying more than 20% of the vehicle's total area. This inferior performance is probably related to the type of dimension (i.e., vertical, homogeneous, or horizontal) of the items, as in 4 out of 7 instances where optimality was not proved, most items were of the homogeneous or horizontal type.

Table 12 provides a detailed analysis of the method's convergence, taking into account the number of customers and items. The instances are grouped first by the number of customers (N_c) and subsequently by the number of items (MI), which are classified into intervals. For each group of instances and solution type (OP or TL), we report the percentage of customers requiring more than one item (NI), the total number of instances (NB_{ins}), the number of customers served with split delivery (SD_C), and the number of instances for which at least one customer received split delivery (SD_{ins}). Additionally, the table presents the average total time required to solve the instances, the average time spent on the packing subproblem, and the gap, defined as the percentage difference between the lower and upper bounds. It is important to note that the gap is not displayed for instances with an optimal solution, as its value is zero in such cases.

As expected, the results in Table 12 demonstrate that as the numbers of customers and items increase, the problem becomes more challenging to solve. For the instances with 35 customers, the method could not find an optimal solution within the time limit for all of them, thus highlighting its limitations. Moreover, in contrast to the other instances for which the method dedicated a considerable amount of time to the packing subproblem, this time was less than 10 s, with the majority of the total time being used for routing. The method is more efficient when solving instances involving many customers who require more than one item. In the group of instances with 32 customers, composed of 5 instances, BC-R2IVF* successfully proved optimality for a single instance with the largest number of items. However, when analyzing the distribution of items' dimensions, we found that only this particular instance, named E033-03n_5 (in class 5, where customers demand a maximum of five items), had the majority of items in the vertical format. In the other instances for which optimality was not proven, most items were either homogeneous or horizontal.

5.5 | Comparison of BC Approaches on the Realistic Instances

We now analyze the results obtained from the realistic instances we introduce in this paper, which belong to classes 7 to 9. The results are summarized in Tables 13–15. We present the detailed results in A, Tables A6–A10. Table 13 has the same structure as Table 6.

The results in Table 13 highlight the superior performance of the BC methods based on the two-index formulation, as similarly observed for the benchmark instances. BC-R2IVF* and BC-R2IVF obtained optimal solutions for more instances than BC-3IVF* and BC-3IVF. They also require less computation time, with an average difference of around 200 s. In addition, the gap values indicate that the lower and upper bounds obtained with BC-3IVF* and BC-3IVF are significantly far from each other, with an average gap greater than 30%. At the same time, we observe gap values less than 4% for BC-R2IVF* and BC-R2IVF. Again, BC-R2IVF* and BC-R2IVF require longer computation times for checking the packing feasibility; however, this is compensated by the shorter time on routing decisions.

Table 14 presents the total number of cuts added in each BC method, following the same nomenclature as in Table 7. We

TABLE 13 | Results on realistic instances.

	BC-3IVF*					BC-3IVF				
Class	OP	TL	T_{total}(s)	T_{pack}(s)	Gap	OP	TL	T_{total}(s)	T_{pack}(s)	Gap
7	2	4	2401.22	371.15	34.72%	2	4	2401.81	465.90	37.49%
8	3	3	1853.88	24.74	20.77%	3	3	1869.71	41.20	40.40%
9	2	4	2533.38	48.62	45.06%	2	4	2474.98	41.26	88.50%
7-9	7	11	2262.83	148.17	33.52%	7	11	2248.83	182.79	55.46%
Class	BC-R2IVF*					BC-R2IVF				
7	3	3	1805.43	1815.53	0.62%	3	3	1832.63	2119.18	1.48%
8	3	3	1800.54	408.95	2.77%	3	3	1802.70	486.80	3.28%
9	2	4	2400.00	541.32	5.34%	2	4	2400.07	943.71	6.03%
7-9	8	10	2001.99	921.93	2.91%	8	10	2011.80	1183.23	3.60%

TABLE 14 | Number of cuts on the solutions of the realistic instances.

Class	BC-3IVF*		BC-3IVF		BC-R2IVF*			BC-R2IVF		
	Cut_E	Cut_P	Cut_E	Cut_P	Cut_R	Cut_P	Cut_S	Cut_R	Cut_P	Cut_S
7	9109	174	8958	170	3563	256	392	3329	279	334
8	3696	35	6650	26	5096	231	783	5994	343	715
9	7795	26	10 504	34	5740	2450	3135	4759	1784	2476
7-9	20 600	235	26 112	230	14 399	2937	4310	14 082	2406	3525

TABLE 15 | Results obtained by the BC methods for the realistic instances.

Instances		BC-R2IVF*	BC-R2IVF	BC-3IVF*	BC-3IVF	$Gap_{BC\text{-}R2IVF}$	$Gap_{BC\text{-}3IVF^*}$	$Gap_{BC\text{-}3IVF}$
Name	Cl							
inst_11	7	534	534	534	534	0.00%	0.00%	0.00%
inst_16	7	670	670	670	670	0.00%	0.00%	0.00%
inst_21	7	810	815	819	815	0.62%	1.11%	0.62%
inst_26	7	953	989	1169	1030	3.78%	22.67%	8.08%
inst_31	7	990	990	1097	1328	0.00%	10.81%	34.14%
inst_36	7	1236	1236	1783	1747	0.00%	44.26%	41.34%
Avg.						0.73%	13.14%	14.03%
inst_11	8	547	547	547	547	0.00%	0.00%	0.00%
inst_16	8	758	758	758	758	0.00%	0.00%	0.00%
inst_21	8	752	752	752	752	0.00%	0.00%	0.00%
inst_26	8	978	978	1017	1056	0.00%	3.99%	7.98%
inst_31	8	1046	1051	1069	1403	0.48%	2.20%	34.13%
inst_36	8	1163	1191	1201	1633	2.41%	3.27%	40.41%
Avg.						0.48%	1.58%	13.75%
inst_11	9	516	516	516	516	0.00%	0.00%	0.00%
inst_16	9	641	641	641	641	0.00%	0.00%	0.00%
inst_21	9	998	993	958	973	-0.50%	-4.01%	-2.51%
inst_26	9	1078	1078	1173	1155	0.00%	8.81%	7.14%
inst_31	9	1072	1120	1215	2029	4.48%	13.34%	89.27%
inst_36	9	1347	1391	1577	2583	3.27%	17.07%	91.76%
Avg.						1.21%	5.87%	30.94%
Avg. 7-9						0.81%	6.86%	19.58%

observe that the number of routes with an infeasible packing is higher in BC-2IVF* and BC-R2IVF than in BC-3IVF* and BC-3IVF. However, more cuts concerning subtour elimination and connectivity of routes are inserted in BC-3IVF* and BC-3IVF compared to BC-R2IVF* and BC-R2IVF.

Finally, Table 15 compares the solutions obtained by BC-R2IVF* with those obtained by the other methods. For each instance, we present the solution value according to each BC method and the gap between the solution of BC-R2IVF* and those of BC-R2IVF, BC-3IVF*, and BC-3IVF. For example, column $Gap_{BC\text{-}R2IVF}$ reports the gap between the solutions obtained by BC-R2IVF* and BC-R2IVF. A positive gap value indicates that the solution obtained with BC-R2IVF* is better. The results show the superior performance of BC-R2IVF*, as it obtains solutions with average relative differences of 0.81%, 6.86%, and 19.58% compared to the solutions obtained by BC-R2IVF, BC-3IVF*, and BC-3IVF, respectively. For only one instance, namely `inst_21` of class 9, BC-R2IVF* performs worse than all the other BC methods. For all other instances, BC-R2IVF* obtain the same or a better solution. It is worth mentioning that the improvement in the solution value obtained by BC-R2IVF* concerning BC-3IVF and BC-3IVF* is greater than 14% on several instances.

6 | Concluding Remarks

We addressed the split delivery vehicle routing problem with two-dimensional loading constraints (2L-SDVRP). This problem aims to minimize transportation costs while determining routes to deliver rectangular items requested by the customers. Split deliveries allow a customer to be visited by two or more vehicles if this reduces the overall cost. Two formulations are presented and discussed: a three-index vehicle flow formulation and a relaxed two-index vehicle flow formulation. They are approached with tailored branch-and-cut (BC) methods, which embed procedures to detect on-the-fly violated constraints due to packing, subtour, and connectivity of routes. Different procedures, including heuristics, lower bounds, and constraint programming models, are used to check whether the packing subproblem of each route is feasible. We also propose a cut to eliminate infeasible solutions of the relaxed two-index vehicle flow formulation.

The BC methods are tested on different instances to analyze their performance regarding solution quality and computation time. The results indicate the superior performance of the methods based on the two-index formulation regarding both solution quality and computation time.

These methods attain more feasible and optimal solutions on the benchmark instances of the 2L-SDVRP. We observe similar conclusions when solving new realistic instances based on real data from Brazilian logistics companies. We believe they bring practical and relevant insights to evaluate the performance of the proposed approaches.

Future research may propose valid inequalities for the packing subproblem, which could be used in the BC methods to find improved solutions and better bounds. Another interesting research topic would be to propose a branch-price-and-cut method, including a column-generation algorithm for handling routing and packing decisions simultaneously. This method has successfully solved VRP variants with split deliveries [5, 34]. Another direction would be to integrate new constraints into the packing subproblem, for example, cargo stability, load balancing, and manual loading, among others, to approximate the problem faced in real situations [69, 70].

Acknowledgments

The authors are thankful to the two anonymous reviewers whose comments helped improve an earlier version of this paper. The authors would also like to thank the support of the São Paulo Research Foundation (FAPESP) (grant numbers 2022/05803-3, 2022/10303-0, and 2023/07988-3); the State of Goiás Research Foundation (FAPEG); and the National Council for Scientific and Technological Development (CNPq) (grant numbers 405369/2021-2, 405702/2021-3, 408722/2023-1, 314079/2023-8, 315555/2023-8, 304618/2023-3, 444679/2024-3, and 407005/2024-2). The Article Processing Charge for the publication of this research was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) (ROR identifier: 00x0ma614).

Data Availability Statement

The data that supports the findings of this study are available upon request from the authors.

References

1. P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications* (SIAM, 2014).

2. M. Iori, J.-J. Salazar-González, and D. Vigo, "An Exact Approach for the Vehicle Routing Problem With Two-Dimensional Loading Constraints," *Transportation Science* 41, no. 2 (2007): 253–264.

3. B. Ji, S. Zhou, S. Y. Samson, and G. Wu, "An Enhanced Neighborhood Search Algorithm for Solving the Split Delivery Vehicle Routing Problem With Two-Dimensional Loading Constraints," *Computers & Industrial Engineering* 162 (2021): 1–11.

4. T. Vidal, G. Laporte, and P. Matl, "A Concise Guide to Existing and Emerging Vehicle Routing Problem Variants," *European Journal of Operational Research* 286, no. 2 (2020): 401–416.

5. I. Balster, T. Bulhões, P. Munari, A. A. Pessoa, and R. Sadykov, "A New Family of Route Formulations for Split Delivery Vehicle Routing Problems," *Transportation Science* 57, no. 5 (2023): 1359–1378.

6. O. X. Nascimento, T. A. Queiroz, and L. Junqueira, "Practical Constraints in the Container Loading Problem: Comprehensive Formulations and Exact Algorithm," *Computers & Operations Research* 128 (2021): 105186.

7. K. M. Ferreira, T. A. Queiroz, and F. M. B. Toledo, "An Exact Approach for the Green Vehicle Routing Problem With Two-Dimensional Loading Constraints and Split Delivery," *Computers & Operations Research* 136 (2021): 1–27.

8. J.-M. Belenguer, M. Martinez, and E. Mota, "A Lower Bound for the Split Delivery Vehicle Routing Problem," *Operations Research* 48, no. 5 (2000): 801–810.

9. C. Archetti, N. Bianchessi, and M. G. Speranza, "Branch-and-Cut Algorithms for the Split Delivery Vehicle Routing Problem," *European Journal of Operational Research* 238, no. 3 (2014): 685–698.

10. P. Munari and M. Savelsbergh, "Compact Formulations for Split Delivery Routing Problems," *Transportation Science* 56, no. 4 (2022): 1022–1043.

11. J. Lysgaard, A. N. Letchford, and R. W. Eglese, "A New Branch-And-Cut Algorithm for the Capacitated Vehicle Routing Problem," *Mathematical Programming* 100 (2004): 423–445.

12. M. Dror and P. Trudeau, "Savings by Split Delivery Routing," *Transportation Science* 23, no. 2 (1989): 141–145.

13. M. Dror and P. Trudeau, "Split Delivery Routing," *Naval Research Logistics* 37, no. 3 (1990): 383–402.

14. C. Archetti and M. G. Speranza, "The Split Delivery Vehicle Routing Problem: A Survey," in *The Vehicle Routing Problem: Latest Advances and New Challenges*, eds. B. Golden, R. Raghavan, and E. Wasil (Springer, 2008), 103–122.

15. C. Archetti and M. G. Speranza, "Vehicle Routing Problems With Split Deliveries," *International Transactions in Operational Research* 19, no. 1–2 (2012): 3–22.

16. M. Dror, G. Laporte, and P. Trudeau, "Vehicle Routing With Split Deliveries," *Discrete Applied Mathematics* 50, no. 3 (1994): 239–254.

17. C.-G. Lee, M. A. Epelman, C. C. White, III, and Y. A. Bozer, "A Shortest Path Approach to the Multiple-Vehicle Routing Problem With Split Pick-Ups," *Transportation Research Part B: Methodological* 40, no. 4 (2006): 265–284.

18. M. Jin, K. Liu, and R. O. Bowden, "A Two-Stage Algorithm With Valid Inequalities for the Split Delivery Vehicle Routing Problem," *International Journal of Production Economics* 105, no. 1 (2007): 228–242.

19. L. Moreno, M. P. De Aragao, and E. Uchoa, "Improved Lower Bounds for the Split Delivery Vehicle Routing Problem," *Operations Research Letters* 38, no. 4 (2010): 302–306.

20. G. Ozbaygin, O. Karasan, and H. Yaman, "New Exact Solution Approaches for the Split Delivery Vehicle Routing Problem," *EURO Journal on Computational Optimization* 6, no. 1 (2018): 85–115.

21. C. Archetti, N. Bianchessi, and M. G. Speranza, "A Column Generation Approach for the Split Delivery Vehicle Routing Problem," *Networks* 58, no. 4 (2011): 241–254.

22. U. Derigs, B. Li, and U. Vogel, "Local Search-Based Metaheuristics for the Split Delivery Vehicle Routing Problem," *Journal of the Operational Research Society* 61, no. 9 (2010): 1356–1364.

23. M. Jin, K. Liu, and B. Eksioglu, "A Column Generation Approach for the Split Delivery Vehicle Routing Problem," *Operations Research Letters* 36, no. 2 (2008): 265–270.

24. M. M. Silva, A. Subramanian, and L. S. Ochi, "An Iterated Local Search Heuristic for the Split Delivery Vehicle Routing Problem," *Computers & Operations Research* 53 (2015): 234–249.

25. C. Archetti, M. G. Speranza, and M. W. Savelsbergh, "An Optimization-Based Heuristic for the Split Delivery Vehicle Routing Problem," *Transportation Science* 42, no. 1 (2008): 22–31.

26. S. Chen, B. Golden, and E. Wasil, "The Split Delivery Vehicle Routing Problem: Applications, Algorithms, Test Problems, and Computational Results," *Networks* 49, no. 4 (2007): 318–329.

27. P. Belfiore and H. T. Y. Yoshizaki, "Scatter Search for a Real-Life Heterogeneous Fleet Vehicle Routing Problem With Time Windows and Split Deliveries in Brazil," *European Journal of Operational Research* 199, no. 3 (2009): 750–758.

28. G. Desaulniers, "Branch-and-Price-and-Cut for the Split-Delivery Vehicle Routing Problem With Time Windows," *Operations Research* 58, no. 1 (2010): 179–192.

29. N. Bianchessi and S. Irnich, "Branch-and-Cut for the Split Delivery Vehicle Routing Problem With Time Windows," *Transportation Science* 53 (2019): 442–462.

30. P. Munari and M. Savelsbergh, *A Column Generation-Based Heuristic for the Split Delivery Vehicle Routing Problem With Time Windows. SN Operations Research Forum, 1* (Springer, 2020), 1–24.

31. M. Rajaei, G. Moslehi, and M. Reisi-Nafchi, "The Split Heterogeneous Vehicle Routing Problem With Three-Dimensional Loading Constraints on a Large Scale," *European Journal of Operational Research* 299, no. 2 (2022): 706–721.

32. M. R. S. Matos, Y. Frota, and L. S. Ochi, "Green Vehicle Routing and Scheduling Problem With Split Delivery," *Electronic Notes in Discrete Mathematics* 69 (2018): 13–20.

33. H. Lei, G. Laporte, and B. Guo, "The Vehicle Routing Problem With Stochastic Demands and Split Deliveries," *INFOR: Information Systems and Operational Research* 50, no. 2 (2012): 59–71.

34. C. Archetti, N. Bianchessi, and M. G. Speranza, "A Branch-Price-And-Cut Algorithm for the Commodity Constrained Split Delivery Vehicle Routing Problem," *Computers & Operations Research* 64 (2015): 1–10.

35. C. Archetti, A. M. Campbell, and M. G. Speranza, "Multicommodity vs. Single-Commodity Routing," *Transportation Science* 50, no. 2 (2016): 461–472.

36. B. Vornhusen and H. Kopfer, "Emission Vehicle Routing Problem With Split Delivery and a Heterogeneous Vehicle Fleet," in *International Conference on Computational Logistics* (Springer, 2015), 76–90.

37. M. Iori and S. Martello, "An Annotated Bibliography of Combined Routing and Loading Problems," *Yugoslav Journal of Operations Research* 23, no. 3 (2013): 311–326.

38. H. Pollaris, K. Braekers, A. Caris, G. K. Janssens, and S. Limbourg, "Vehicle Routing Problems With Loading Constraints: State-Of-The-Art and Future Directions," *OR Spectrum* 37, no. 2 (2015): 297–330.

39. P. Hokama, F. K. Miyazawa, and E. C. Xavier, "A Branch-And-Cut Approach for the Vehicle Routing Problem With Loading Constraints," *Expert Systems With Applications* 47 (2016): 1–13.

40. J.-F. Côté, G. Guastaroba, and M. G. Speranza, "The Value of Integrating Loading and Routing," *European Journal of Operational Research* 257, no. 1 (2017): 89–105.

41. L. C. D. Silva, T. A. Queiroz, and F. M. B. Toledo, "Integer Formulations for the Integrated Vehicle Routing Problem With Two-Dimensional Packing Constraints," *Pesquisa Operacional* 42 (2022): e248686, https://doi.org/10.1590/0101-7438.2022.042.00248686.

42. X. Zhang, L. Chen, M. Gendreau, and A. Langevin, "A Branch-And-Cut Algorithm for the Vehicle Routing Problem With Two-Dimensional Loading Constraints," *European Journal of Operational Research* 302, no. 1 (2022): 259–269.

43. X. Zhang, L. Chen, M. Gendreau, and A. Langevin, "A Branch-And-Price-And-Cut Algorithm for the Vehicle Routing Problem With Two-Dimensional Loading Constraints," *Transportation Science* 56, no. 6 (2022): 1618–1635.

44. M. Gendreau, M. Iori, G. Laporte, and S. Martello, "A Tabu Search Heuristic for the Vehicle Routing Problem With Two-Dimensional Loading Constraints," *Networks* 51, no. 1 (2008): 4–18.

45. E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "A Guided Tabu Search for the Vehicle Routing Problem With Two-Dimensional Loading Constraints," *European Journal of Operational Research* 195, no. 3 (2009): 729–743.

46. S. C. Leung, X. Zhou, D. Zhang, and J. Zheng, "Extended Guided Tabu Search and a New Packing Algorithm for the Two-Dimensional Loading Vehicle Routing Problem," *Computers & Operations Research* 38, no. 1 (2011): 205–215.

47. L. Wei, Z. Zhang, D. Zhang, and A. Lim, "A Variable Neighborhood Search for the Capacitated Vehicle Routing Problem With Two-Dimensional Loading Constraints," *European Journal of Operational Research* 243, no. 3 (2015): 798–814.

48. E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "Integrated Distribution and Loading Planning via a Compact Metaheuristic Algorithm," *European Journal of Operational Research* 228, no. 1 (2013): 56–71.

49. S. C. Leung, J. Zheng, D. Zhang, and X. Zhou, "Simulated Annealing for the Vehicle Routing Problem With Two-Dimensional Loading Constraints," *Flexible Services and Manufacturing Journal* 22, no. 1–2 (2010): 61–82.

50. L. Wei, Z. Zhang, D. Zhang, and S. C. Leung, "A Simulated Annealing Algorithm for the Capacitated Vehicle Routing Problem With Two-Dimensional Loading Constraints," *European Journal of Operational Research* 265, no. 3 (2018): 843–859.

51. X. Song, D. Jones, N. Asgari, and T. Pigden, "Multi-Objective Vehicle Routing and Loading With Time Window Constraints: A Real-Life Application," *Annals of Operations Research* 291 (2020): 799–825.

52. K. M. Ferreira and T. A. Queiroz, "A Simulated Annealing Based Heuristic for a Location-Routing Problem With Two-Dimensional Loading Constraints," *Applied Soft Computing* 118 (2022): 108443.

53. E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "Vehicle Routing Strategies for Pick-Up and Delivery Service Under Two Dimensional Loading Constraints," *Operational Research* 17 (2017): 115–143.

54. O. Dominguez, A. A. Juan, B. Barrios, J. Faulin, and A. Agustin, "Using Biased Randomization for Solving the Two-Dimensional Loading Vehicle Routing Problem With Heterogeneous Fleet," *Annals of Operations Research* 236 (2016): 383–404.

55. S. C. Leung, Z. Zhang, D. Zhang, X. Hua, and M. K. Lim, "A Meta-Heuristic Algorithm for Heterogeneous Fleet Vehicle Routing Problems With Two-Dimensional Loading Constraints," *European Journal of Operational Research* 225, no. 2 (2013): 199–210.

56. O. Dominguez, D. Guimarans, A. A. Juan, and I. de la Nuez, "A Biased-Randomised Large Neighbourhood Search for the

Two-Dimensional Vehicle Routing Problem With Backhauls," *European Journal of Operational Research* 255, no. 2 (2016): 442–462.

57. D. Guimarans, O. Dominguez, J. Panadero, and A. A. Juan, "A Simheuristic Approach for the Two-Dimensional Vehicle Routing Problem With Stochastic Travel Times," *Simulation Modelling Practice and Theory* 89 (2018): 1–14.

58. J.-F. Côté, M. Gendreau, and J.-Y. Potvin, "The Vehicle Routing Problem With Stochastic Two-Dimensional Items," *Transportation Science* 54, no. 2 (2020): 453–469.

59. S. Ceschia, A. Schaerf, and T. Stützle, "Local Search Techniques for a Routing-Packing Problem," *Computers & Industrial Engineering* 66, no. 4 (2013): 1138–1149.

60. X. Li, M. Yuan, D. Chen, J. Yao, and J. Zeng, "A Data-Driven Three-Layer Algorithm for Split Delivery Vehicle Routing Problem With 3D Container Loading Constraint," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Association for Computing Machinery, 2018), 528–536.

61. J. Yi and A. Bortfeldt, "The Capacitated Vehicle Routing Problem With Three-Dimensional Loading Constraints and Split Delivery—A Case Study," in *Operations Research Proceedings 2016* (Springer, 2018), 351–356.

62. A. Bortfeldt and J. Yi, "The Split Delivery Vehicle Routing Problem With Three-Dimensional Loading Constraints," *European Journal of Operational Research* 282, no. 2 (2020): 545–558.

63. J. Herz, "Recursive Computational Procedure for Two-Dimensional Stock Cutting," *IBM Journal of Research and Development* 16, no. 5 (1972): 462–469.

64. M. A. Boschetti and L. Montaletti, "An Exact Algorithm for the Two-Dimensional Strip-Packing Problem," *Operations Research* 58, no. 6 (2010): 1774–1791.

65. J.-F. Côté, M. Gendreau, and J.-Y. Potvin, "An Exact Algorithm for the Two-Dimensional Orthogonal Packing Problem With Unloading Constraints," *Operations Research* 62, no. 5 (2014): 1126–1141.

66. F. Clautiaux, A. Jouglet, J. Carlier, and A. Moukrim, "A New Constraint Programming Approach for the Orthogonal Packing Problem," *Computers & Operations Research* 35, no. 3 (2008): 944–959.

67. R. W. Floyd, "Algorithm 97: Shortest Path," *Communications of the ACM* 5, no. 6 (1962): 345.

68. R. Morabito, S. R. Morales, and J. A. Widmer, "Loading Optimization of Palletized Products on Trucks," *Transportation Research Part E: Logistics and Transportation Review* 36, no. 4 (2000): 285–296.

69. L. de Azevedo Oliveira, V. L. de Lima, T. A. Queiroz, and F. K. Miyazawa, "The Container Loading Problem With Cargo Stability: A Study on Support Factors, Mechanical Equilibrium and Grids," *Engineering Optimization* 53, no. 7 (2021): 1192–1211.

70. L. Junqueira and T. A. Queiroz, "The Static Stability of Support Factor-Based Rectangular Packings: An Assessment by Regression Analysis," *International Transactions in Operational Research* 29, no. 1 (2022): 574–599.

Appendix A

Detailed Results

In this appendix, we report detailed results obtained by the BC methods. Results related to the benchmark instances (classes 2 to 6) are reported in Tables A1–A5. For the newly generated instances (classes 7 to 9), results are reported in Tables A6–A10. All these tables report the following information for each instance: name, class, the number of customers (n), the total number of items ($|R_j|$), the number of vehicles (K), the solution cost (Obj); the total computation time in seconds (Time_T); the computation time for solving the packing subproblems in seconds (Time_P); the number of capacity and connectivity cuts (Cut_R); and the number of cuts concerning routes with infeasible packing (Cut_P). In addition, Tables A2 and A10 present the minimum number of vehicles needed to serve all customers ($K_{\min}$), the number of vehicles in the resulting solution (VH), the number of customers served with split delivery (CS). For BC-R2IVF* and BC-R2IVF, we also show the number of cuts regarding the residual optimization problem (Cut_S). The letters "tl" indicate that the imposed time limit was reached for the instance.

TABLE A1 | Information of the initial solutions used in the BC methods on benchmark instances.

Instances					Solution				
Name	Class	n	$\lvert R_j \rvert$	K	Obj	Time_T	Time_P	Cut_R	Cut_P
E016-03m	2	15	24	3	285	2.80	0.76	949	32
	3	15	31	3	280	8.50	8.07	513	10
	4	15	37	4	288	0.79	0.77	43	4
	5	15	45	4	279	1.46	1.45	23	2
	6	15	48	4	277	0.20	0.19	11	0
E016-05m	2	15	25	5	342	0.81	0.03	654	5
	3	15	31	5	347	2.52	0.30	1233	7
	4	15	40	5	336	1.18	0.91	326	2
	5	15	48	5	329	0.25	0.15	198	0
	6	15	48	5	329	0.10	0.00	198	0
E021-04m	2	20	29	5	395	15.98	2.75	2853	396
	3	20	46	5	387	2.00	1.65	388	7
	4	20	44	5	374	10.10	9.78	180	7
	5	20	49	5	369	0.72	0.48	120	0
	6	20	66	5	369	0.96	0.85	121	0
E021-06m	2	20	32	6	434	2.43	0.22	1649	13
	3	20	43	6	432	2.14	1.62	575	5
	4	20	50	6	438	5.50	2.55	1433	1
	5	20	62	6	423	0.08	0.00	305	0
	6	20	66	6	423	0.27	0.11	388	0
E022-04g	2	21	31	4	379	4.01	3.65	454	24
	3	21	37	4	373	0.35	0.29	281	3
	4	21	41	4	377	0.31	0.19	312	2
	5	21	57	5	388	0.36	0.29	91	0
	6	21	68	5	388	0.35	0.32	85	0
E022-06m	2	21	33	6	490	1.16	0.08	994	6
	3	21	40	6	495	6.47	3.97	1310	20
	4	21	57	6	488	0.37	0.14	618	1
	5	21	56	6	487	0.33	0.09	562	0
	6	21	68	6	487	0.65	0.09	587	0
E023-03g	2	22	32	5	723	2.21	1.51	428	28
	3	22	41	5	698	12.04	11.82	193	42
	4	22	51	5	714	10.11	9.99	169	7
	5	22	55	6	742	3.14	3.10	23	1
	6	22	70	6	745	13.04	12.99	39	0
E023-05s	2	22	29	5	720	9.69	8.41	410	301
	3	22	42	5	730	9.71	9.08	409	50
	4	22	48	5	701	1.21	1.16	82	2
	5	22	52	6	720	0.14	0.12	21	0
	6	22	70	6	745	12.81	12.76	39	0
E026-08m	2	25	40	8	612	1.63	0.08	1852	4
	3	25	61	8	615	1.65	0.25	1077	2
	4	25	63	8	625	24.96	0.31	3238	6
	5	25	91	8	609	0.27	0.05	417	0
	6	25	79	8	609	0.34	0.00	619	0
E030-03g	2	29	43	6	684	33.98	11.50	3695	801
	3	29	49	6	635	69.77	52.45	2291	231
	4	29	72	7	733	43.80	43.54	369	35
	5	29	86	7	702	52.12	51.34	449	4
	6	29	91	7	688	0.61	0.40	192	0
E033-03n	2	32	44	7	2713	300.00	14.04	8718	1395
	3	32	56	7	2574	300.00	95.59	3021	312
	4	32	78	7	2667	300.00	238.54	3383	146
	5	32	102	8	2631	105.59	105.15	112	7
	6	32	99	8	2628	1.25	1.15	73	0
E036-11h	2	35	56	11	682	169.07	0.07	6833	0
	3	35	74	11	682	163.62	0.14	7295	0
	4	35	93	11	690	300.00	13.39	15 985	19
	5	35	114	11	682	227.52	0.21	8339	0
	6	35	109	11	682	280.64	0.00	9222	0

TABLE A2 | Results of the BC-R2IVF* on benchmark instances.

Instances				Solution									
Name	Class	n	$\|R_j\|$	K_{min}	K	VH	CS	Obj	$Time_T$	$Time_P$	Cut_R	Cut_P	Cut_S
E016-03m	2	15	24	3	3	3	1	279	30.73	4.86	681	176	280
	3	15	31	3	3	3	0	280	57.95	26.86	576	122	225
	4	15	37	3	4	4	0	288	61.62	28.18	979	17	131
	5	15	45	3	4	3	1	262	15.38	13.78	80	2	0
	6	15	48	3	4	3	1	262	0.49	0.11	62	0	0
E016-05m	2	15	25	5	5	5	1	329	268.56	0.33	700	67	1003
	3	15	31	5	5	5	2	329	864.88	1.64	456	39	990
	4	15	40	5	5	5	2	308	1.55	0.14	38	0	0
	5	15	48	5	5	5	1	308	0.51	0.12	32	0	0
	6	15	48	5	5	5	1	308	0.33	0.00	34	0	0
E021-04m	2	20	29	4	5	5	0	395	tl	36.31	2539	1928	3263
	3	20	46	4	5	5	0	387	tl	92.10	1436	500	2147
	4	20	44	4	5	4	2	353	29.25	9.52	397	31	26
	5	20	49	4	5	4	2	348	2.11	0.37	287	0	1
	6	20	66	4	5	4	2	349	616.23	602.73	425	6	22
E021-06m	2	20	32	6	6	6	0	434	551.97	1.87	1027	44	495
	3	20	43	6	6	6	3	419	19.93	1.04	321	4	9
	4	20	50	6	6	6	2	420	92.22	1.10	401	2	25
	5	20	62	6	6	6	3	420	177.52	0.09	304	0	25
	6	20	66	6	6	6	4	416	11.05	0.32	266	0	0
E022-04g	2	21	31	4	4	4	1	376	20.77	5.80	343	37	179
	3	21	37	4	4	4	0	373	2.67	0.34	162	8	34
	4	21	41	4	4	4	0	377	12.67	2.06	215	38	102
	5	21	57	4	5	4	0	367	1.03	0.42	182	0	0
	6	21	68	4	5	4	0	367	1.01	0.43	143	0	0
E022-06m	2	21	33	6	6	6	1	472	10.53	0.21	213	6	8
	3	21	40	6	6	6	1	485	151.60	4.66	533	14	292
	4	21	57	6	6	6	1	470	1.11	0.29	182	0	0
	5	21	56	6	6	6	1	470	1.44	0.13	160	0	2
	6	21	68	6	6	6	2	470	3.65	0.12	130	0	1
E023-03g	2	22	32	4	5	5	0	723	tl	51.78	1157	2024	6894
	3	22	41	4	5	4	1	673	tl	392.95	857	2297	14 023
	4	22	51	4	5	4	1	682	1726.46	909.66	528	501	6430
	5	22	55	3	6	4	0	647	tl	1235.27	846	67	2950
	6	22	70	4	6	4	1	649	58.92	43.38	85	4	1
E023-05s	2	22	29	4	5	5	1	699	tl	71.83	1697	3029	11 724
	3	22	42	4	5	5	0	730	tl	206.44	1167	516	14 589
	4	22	48	4	5	4	1	673	tl	1824.22	397	816	4424
	5	22	52	3	6	4	0	632	tl	3662.95	249	34	160
	6	22	70	4	6	4	1	649	58.53	43.19	85	4	1
E026-08m	2	25	40	8	8	8	0	598	801.47	0.23	587	10	167
	3	25	61	8	8	8	0	601	tl	0.88	635	4	277
	4	25	63	8	8	8	2	621	tl	0.54	727	4	147
	5	25	91	8	8	8	4	585	62.15	0.41	410	0	4
	6	25	79	8	8	8	3	594	tl	0.41	415	0	228
E030-03g	2	29	43	5	6	6	0	684	tl	36.29	1328	1305	2774
	3	29	49	4	6	6	0	635	tl	539.16	1266	1373	1400
	4	29	72	6	7	7	0	733	tl	839.45	720	434	407
	5	29	86	5	7	6	0	681	tl	1812.67	977	42	233
	6	29	91	5	7	6	2	619	tl	1340.97	1070	18	354
E033-03n	2	32	44	5	7	7	0	2713	tl	51.74	6238	3137	1355
	3	32	56	5	7	7	0	2574	tl	284.16	1565	1089	452
	4	32	78	6	7	7	0	2667	tl	1414.43	1829	939	1291
	5	32	102	5	8	5	0	2320	476.01	178.35	1486	4	443
	6	32	99	5	8	5	3	2378	tl	2945.05	1043	74	1067
E036-11h	2	35	56	11	11	11	0	682	tl	0.04	3276	0	28
	3	35	74	11	11	11	4	680	tl	0.26	2754	0	16
	4	35	93	11	11	11	0	690	tl	8.71	3510	11	19
	5	35	114	11	11	11	2	673	tl	0.09	3017	0	24
	6	35	109	11	11	11	0	682	tl	0.04	3715	0	10

TABLE A3 | Results of the BC-R2IVF on benchmark instances.

Instances				Solution									
Name	**Class**	n	$\lvert R_j \rvert$	K_{min}	K	VH	CS	**Obj**	**Time$_T$**	**Time$_P$**	**Cut$_R$**	**Cut$_P$**	**Cut$_S$**
E016-03m	2	15	24	3	3	3	1	279	24.74	5.83	591	165	205
	3	15	31	3	3	3	0	280	81.73	43.18	757	203	395
	4	15	37	3	4	4	0	288	73.58	41.65	1112	27	131
	5	15	45	3	4	3	1	262	15.25	15.13	100	2	1
	6	15	48	3	4	3	1	262	0.65	0.33	121	0	0
E016-05m	2	15	25	5	5	5	1	329	279.11	0.39	512	74	968
	3	15	31	5	5	5	2	329	1078.79	2.30	531	60	1447
	4	15	40	5	5	5	1	308	7.99	0.06	43	0	0
	5	15	48	5	5	5	1	308	2.28	0.00	49	0	1
	6	15	48	5	5	5	1	308	0.83	0.00	30	0	0
E021-04m	2	20	29	4	5	5	2	395	tl	35.62	2598	2339	3582
	3	20	46	4	5	5	1	392	tl	157.01	1322	707	1379
	4	20	44	4	5	4	2	353	73.99	49.03	671	106	42
	5	20	49	4	5	4	2	348	24.29	5.69	360	2	5
	6	20	66	4	5	4	2	349	16.06	5.10	460	1	8
E021-06m	2	20	32	6	6	6	1	434	601.98	1.80	1160	51	505
	3	20	43	6	6	6	3	419	33.03	0.80	414	4	23
	4	20	50	6	6	6	2	420	573.47	55.20	476	121	32
	5	20	62	6	6	6	2	420	317.47	1.32	401	0	38
	6	20	66	6	6	6	2	416	134.19	62.74	411	2	1
E022-04g	2	21	31	4	4	4	1	376	23.28	9.91	556	65	244
	3	21	37	4	4	4	1	373	8.66	0.59	356	12	60
	4	21	41	4	4	4	1	377	14.21	2.27	399	38	141
	5	21	57	4	5	4	0	367	6.69	3.14	211	1	1
	6	21	68	4	5	4	0	367	13.14	0.77	335	0	0
E022-06m	2	21	33	6	6	6	1	472	133.95	0.29	342	38	59
	3	21	40	6	6	6	1	485	298.83	7.77	398	23	235
	4	21	57	6	6	6	2	470	284.46	25.27	270	6	42
	5	21	56	6	6	6	2	470	193.20	0.17	293	0	63
	6	21	68	6	6	6	1	470	41.20	0.05	286	0	9
E023-03g	2	22	32	4	5	5	1	715	tl	57.99	1362	2255	8290
	3	22	41	4	5	5	1	698	tl	659.86	1109	2804	8703
	4	22	51	4	5	4	1	682	1834.59	934.40	568	584	7000
	5	22	55	3	6	4	1	645	tl	1619.65	849	80	2532
	6	22	70	4	6	4	1	649	127.94	122.55	161	1	3
E023-05s	2	22	29	4	5	5	1	699	tl	78.02	1516	3307	11 966
	3	22	42	4	5	5	1	731	tl	227.42	1095	584	11 100
	4	22	48	4	5	4	1	673	tl	2064.97	519	895	4609
	5	22	52	3	6	4	0	633	tl	3526.58	332	32	184
	6	22	70	4	6	4	1	649	129.57	124.25	161	1	3
E026-08m	2	25	40	8	8	8	0	598	1185.39	0.84	676	28	205
	3	25	61	8	8	—	—	—	tl	11.46	0	20	1
	4	25	63	8	8	—	—	—	tl	5.92	0	2	1
	5	25	91	8	8	8	3	585	18.93	0.53	382	0	0
	6	25	79	8	8	—	—	—	tl	0.30	0	0	1
E030-03g	2	29	43	5	6	6	1	826	tl	55.89	721	1502	568
	3	29	49	4	6	6	1	663	tl	606.72	909	1295	504
	4	29	72	6	7	7	2	736	tl	946.59	581	426	304
	5	29	86	5	7	7	3	699	tl	2602.90	548	81	54
	6	29	91	5	7	6	3	619	tl	1993.33	560	24	599
E033-03n	2	32	44	5	7	7	4	3171	tl	18.23	2691	981	119
	3	32	56	5	7	7	2	2725	tl	390.83	1208	1470	237
	4	32	78	6	7	7	1	2635	tl	1327.72	1637	840	866
	5	32	102	5	8	5	0	2320	789.95	601.88	942	4	434
	6	32	99	5	8	5	3	2394	tl	2321.32	1298	38	1882
E036-11h	2	35	56	11	11	—	—	—	tl	0.82	0	6	1
	3	35	74	11	11	—	—	—	tl	0.00	141	0	3
	4	35	93	11	11	—	—	—	tl	0.00	0	0	1
	5	35	114	11	11	11	5	696	tl	0.71	2566	0	5
	6	35	109	11	11	—	—	—	tl	0.46	130	0	1

TABLE A4 | Results of the BC-3IVF* on benchmark instances.

Instances				Solution								
Name	Class	n	$\|R_j\|$	K_{min}	K	VH	CS	Obj	$Time_T$	$Time_P$	Cut_R	Cut_P
E016-03m	2	15	24	3	3	3	1	279	10.65	5.34	120	175
	3	15	31	3	3	3	0	280	30.86	20.36	54	62
	4	15	37	3	4	4	0	288	19.86	9.70	122	5
	5	15	45	3	4	3	1	262	49.21	46.82	60	3
	6	15	48	3	4	3	1	262	1.81	0.64	32	0
E016-05m	2	15	25	5	5	5	1	329	19.84	0.22	102	74
	3	15	31	5	5	5	2	329	21.47	1.08	73	27
	4	15	40	5	5	5	2	308	3.83	0.12	38	0
	5	15	48	5	5	5	2	308	2.96	0.35	16	0
	6	15	48	5	5	5	2	308	2.66	0.09	10	0
E021-04m	2	20	29	4	5	5	0	395	tl	9.31	1052	494
	3	20	46	4	5	5	0	387	tl	8.01	556	57
	4	20	44	4	5	4	3	353	tl	26.27	450	47
	5	20	49	4	5	4	2	348	433.81	0.57	284	0
	6	20	66	4	5	4	2	349	554.20	2.04	212	0
E021-06m	2	20	32	6	6	6	0	434	tl	0.16	282	2
	3	20	43	6	6	6	3	419	tl	0.69	300	4
	4	20	50	6	6	6	3	423	tl	1.26	304	5
	5	20	62	6	6	6	0	423	tl	0.00	250	0
	6	20	66	6	6	6	3	416	tl	0.22	244	0
E022-04g	2	21	31	4	4	4	1	376	69.72	2.86	190	22
	3	21	37	4	4	4	0	373	29.01	0.53	128	8
	4	21	41	4	4	4	0	377	46.05	2.46	206	32
	5	21	57	4	5	4	1	367	51.50	0.61	126	0
	6	21	68	4	5	4	0	367	55.86	12.11	116	0
E022-06m	2	21	33	6	6	6	2	472	tl	0.28	694	3
	3	21	40	6	6	6	4	486	tl	3.41	692	19
	4	21	57	6	6	6	2	470	tl	0.39	370	0
	5	21	56	6	6	6	3	470	tl	0.07	662	0
	6	21	68	6	6	6	3	470	tl	0.27	698	0
E023-03g	2	22	32	4	5	5	0	723	tl	8.19	970	226
	3	22	41	4	5	4	1	673	tl	125.66	1069	391
	4	22	51	4	5	4	1	684	tl	93.33	750	87
	5	22	55	3	6	4	1	645	3413.91	1273.12	528	65
	6	22	70	4	6	4	1	649	1487.23	22.49	353	0
E023-05s	2	22	29	4	5	4	2	680	tl	6.93	1075	287
	3	22	42	4	5	5	0	730	tl	2.94	803	30
	4	22	48	4	5	4	1	686	tl	154.66	881	85
	5	22	52	3	6	3	1	604	tl	3370.88	266	27
	6	22	70	4	6	4	1	649	1490.12	22.74	353	0
E026-08m	2	25	40	8	8	8	2	598	tl	0.04	392	2
	3	25	61	8	8	8	0	601	tl	0.79	280	4
	4	25	63	8	8	8	0	611	tl	0.16	450	0
	5	25	91	8	8	8	0	595	tl	0.20	308	0
	6	25	79	8	8	8	0	595	tl	0.17	336	0
E030-03g	2	29	43	5	6	6	0	684	tl	2.83	3450	64
	3	29	49	4	6	6	0	632	tl	13.80	2998	12
	4	29	72	6	7	7	1	721	tl	91.61	2256	36
	5	29	86	5	7	5	1	691	tl	178.90	2330	5
	6	29	91	5	7	6	3	624	tl	268.01	2042	9
E033-03n	2	32	44	5	7	7	0	2713	tl	0.75	1884	15
	3	32	56	5	7	7	0	2574	tl	14.77	1666	85
	4	32	78	6	7	7	0	2667	tl	36.14	1560	28
	5	32	102	5	8	5	3	2343	tl	951.12	1042	9
	6	32	99	5	8	6	1	2472	tl	1.35	1062	0
E036-11h	2	35	56	11	11	11	0	682	tl	0.06	270	0
	3	35	74	11	11	11	0	682	tl	0.04	198	0
	4	35	93	11	11	11	0	690	tl	0.23	322	0
	5	35	114	11	11	11	0	682	tl	0.11	272	0
	6	35	109	11	11	11	0	682	tl	0.00	396	0

TABLE A5 | Results of the BC-3IVF on benchmark instances.

Instances				Solution								
Name	Class	n	$\|R_j\|$	K_{min}	K	VH	CS	Obj	Time_T	Time_P	Cut_R	Cut_P
E016-03m	2	15	24	3	3	3	1	279	12.03	6.52	162	172
	3	15	31	3	3	3	1	280	38.90	33.41	150	85
	4	15	37	3	4	4	0	288	49.42	40.10	160	37
	5	15	45	3	4	3	1	262	2.46	0.92	78	0
	6	15	48	3	4	3	1	262	1.97	0.81	52	0
E016-05m	2	15	25	5	5	5	1	329	73.14	0.36	162	69
	3	15	31	5	5	5	2	329	51.99	1.92	105	51
	4	15	40	5	5	5	2	308	4.09	0.23	28	1
	5	15	48	5	5	5	2	308	3.59	0.53	30	0
	6	15	48	5	5	5	2	308	5.18	0.12	32	0
E021-04m	2	20	29	4	5	5	3	431	tl	26.45	2818	1657
	3	20	46	4	5	5	1	398	tl	94.53	1730	360
	4	20	44	4	5	4	2	353	3398.21	76.60	538	99
	5	20	49	4	5	4	2	348	1349.43	1.09	394	0
	6	20	66	4	5	4	3	349	1052.42	635.68	318	6
E021-06m	2	20	32	6	6	6	3	434	tl	2.57	866	61
	3	20	43	6	6	6	4	419	tl	1.37	676	3
	4	20	50	6	6	6	3	424	tl	17.15	764	26
	5	20	62	6	6	6	1	423	tl	1.47	462	0
	6	20	66	6	6	6	4	419	tl	1.20	506	0
E022-04g	2	21	31	4	4	4	1	376	39.02	5.37	304	43
	3	21	37	4	4	4	1	373	45.11	1.73	306	35
	4	21	41	4	4	4	0	377	141.75	34.26	594	114
	5	21	57	4	5	4	0	367	88.67	7.55	284	0
	6	21	68	4	5	4	0	367	3600.00	3490.04	939	4
E022-06m	2	21	33	6	6	6	3	472	3263.41	0.43	784	9
	3	21	40	6	6	6	3	487	tl	6.83	866	29
	4	21	57	6	6	6	4	470	tl	30.96	976	43
	5	21	56	6	6	6	3	470	tl	1.47	762	0
	6	21	68	6	6	6	3	472	tl	1.38	748	0
E023-03g	2	22	32	4	5	5	0	833	tl	39.73	3241	1317
	3	22	41	4	5	4	2	685	tl	151.72	1964	415
	4	22	51	4	5	4	1	684	tl	128.05	1092	108
	5	22	55	3	6	4	1	645	3553.20	1241.07	881	66
	6	22	70	4	6	4	1	649	2161.60	574.34	665	20
E023-05s	2	22	29	4	5	5	0	731	tl	19.42	2886	894
	3	22	42	4	5	5	1	735	tl	26.19	1327	114
	4	22	48	4	5	4	1	673	tl	366.52	959	170
	5	22	52	3	6	3	1	611	tl	3649.97	458	42
	6	22	70	4	6	4	1	649	2163.79	576.51	665	20
E026-08m	2	25	40	8	8	8	5	669	tl	2.59	1430	60
	3	25	61	8	8	8	7	692	tl	1.77	1134	9
	4	25	63	8	8	8	4	680	tl	0.71	648	0
	5	25	91	8	8	8	6	678	tl	1.66	900	0
	6	25	79	8	8	8	4	637	tl	0.27	530	0
E030-03g	2	29	43	5	6	6	—	—	tl	10.87	8460	242
	3	29	49	4	6	6	5	1091	tl	45.44	7860	133
	4	29	72	6	7	7	—	—	tl	138.22	5032	57
	5	29	86	5	7	6	2	737	tl	352.79	2900	10
	6	29	91	5	7	6	5	666	tl	1268.33	2460	15
E033-03n	2	32	44	5	7	6	—	—	tl	6.66	5706	217
	3	32	56	5	7	—	—	—	tl	37.53	4876	179
	4	32	78	6	7	—	—	—	tl	85.97	17 071	99
	5	32	102	5	8	6	4	2565	tl	1236.78	15 355	15
	6	32	99	5	8	6	5	3367	tl	234.72	2889	13
E036-11h	2	35	56	11	11	11	—	—	tl	0.22	848	2
	3	35	74	11	11	11	—	—	tl	0.32	8407	1
	4	35	93	11	11	11	—	—	tl	3.76	490	1
	5	35	114	11	11	11	5	887	tl	1.82	936	0
	6	35	109	11	11	11	—	—	tl	0.06	622	0

TABLE A6 | Information of the initial solutions used in the BC methods on newly generated instances.

Instances					Solution				
Name	Class	n	$\lvert R_j \rvert$	K	Obj	Time_T	Time_P	Cut_R	Cut_P
inst_11	7	10	68	4	770	0.03	0.00	4	0
	8	10	68	4	759	0.00	0.00	0	0
	9	10	60	3	611	0.00	0.00	6	0
inst_16	7	15	90	5	1027	0.00	0.00	2	0
	8	15	122	6	1003	0.13	0.13	6	0
	9	15	106	5	759	0.01	0.00	7	0
inst_21	7	20	140	7	1121	0.01	0.00	4	0
	8	20	136	7	1060	0.06	0.00	10	0
	9	20	152	7	1247	0.49	0.44	52	3
inst_26	7	25	162	8	1513	12.64	12.53	33	3
	8	25	184	9	1206	0.49	0.47	24	0
	9	25	178	9	1433	0.43	0.41	33	0
inst_31	7	30	196	9	1401	10.15	10.05	78	1
	8	30	218	10	1362	0.04	0.00	30	0
	9	30	206	10	1535	0.06	0.00	54	0
inst_36	7	35	272	13	1841	0.03	0.00	38	0
	8	35	242	11	1389	0.83	0.17	311	8
	9	35	256	12	1719	0.37	0.28	66	3

TABLE A7 | Results of the BC-R2IVF* on newly generated instances.

Instances				Solution									
Name	Class	n	$\lvert R_j \rvert$	K_{min}	K	VH	CS	Obj	Time_T	Time_P	Cut_R	Cut_P	Cut_S
inst_11	7	10	68	2	4	2	0	534	0.23	0.14	13	0	0
	8	10	68	2	4	2	0	547	0.19	0.04	32	0	4
	9	10	60	2	3	2	0	516	0.02	0.00	21	0	0
inst_16	7	15	90	2	5	2	0	670	1.56	0.98	86	0	0
	8	15	122	3	6	3	0	758	2.32	0.39	72	4	6
	9	15	106	3	5	3	0	641	0.65	0.00	71	0	1
inst_21	7	20	140	3	7	3	1	810	tl	4116.79	256	116	58
	8	20	136	3	7	3	0	752	1.49	1.03	117	2	1
	9	20	152	4	7	5	1	998	tl	1302.20	936	1159	2068
inst_26	7	25	162	4	8	4	1	953	tl	3516.41	359	132	84
	8	25	184	4	9	5	1	978	tl	9.63	773	53	603
	9	25	178	4	9	5	2	1078	tl	613.10	1480	579	804
inst_31	7	30	196	4	9	4	0	990	tl	3247.76	917	8	249
	8	30	218	5	10	6	1	1046	tl	1787.02	608	39	146
	9	30	206	5	10	6	1	1072	tl	45.43	929	303	255
inst_36	7	35	272	6	13	6	0	1236	30.79	11.12	1932	0	1
	8	35	242	5	11	6	1	1163	tl	655.59	3494	133	23
	9	35	256	6	12	8	4	1347	tl	1287.21	2303	409	7

TABLE A8 | Results of the BC-R2IVF on newly generated instances.

Instances				Solution									
Name	Class	n	$\|R_j\|$	K_{min}	K	VH	CS	Obj	Time_T	Time_P	Cut_R	Cut_P	Cut_S
inst_11	7	10	68	2	4	2	0	534	0.18	0.17	36	0	0
	8	10	68	2	4	2	0	547	0.20	0.00	62	0	6
	9	10	60	2	3	2	0	516	0.01	0.00	26	0	0
inst_16	7	15	90	2	5	2	0	670	2.58	1.27	91	0	0
	8	15	122	3	6	3	0	758	3.41	0.32	97	4	10
	9	15	106	3	5	3	1	641	0.40	0.00	108	0	0
inst_21	7	20	140	3	7	3	0	815	tl	5651.54	303	132	46
	8	20	136	3	7	3	1	752	12.61	10.96	173	6	2
	9	20	152	4	7	5	1	993	tl	2057.81	752	771	1592
inst_26	7	25	162	4	8	5	0	989	tl	3646.36	371	135	71
	8	25	184	4	9	5	2	978	tl	304.37	1207	111	538
	9	25	178	4	9	5	1	1078	tl	1877.89	936	482	732
inst_31	7	30	196	4	9	4	1	990	tl	3376.46	647	8	216
	8	30	218	5	10	6	1	1051	tl	1772.43	804	36	143
	9	30	206	5	10	6	0	1120	tl	1215.93	790	416	144
inst_36	7	35	272	6	13	6	3	1236	193.03	39.27	1881	4	1
	8	35	242	5	11	6	1	1191	tl	832.71	3651	186	16
	9	35	256	6	12	8	1	1391	tl	510.65	2147	115	8

TABLE A9 | Results of the BC-3IVF* on newly generated instances.

Instances				Solution								
Name	Class	n	$\|R_j\|$	K_{min}	K	VH	CS	Obj	Time_T	Time_P	Cut_R	Cut_P
inst_11	7	10	68	2	4	2	0	534	1.61	1.36	14	0
	8	10	68	2	4	2	0	547	0.11	0.00	16	0
	9	10	60	2	3	2	0	516	0.13	0.00	18	0
inst_16	7	15	90	2	5	2	0	670	5.72	1.03	186	0
	8	15	122	3	6	3	0	758	21.61	1.07	142	8
	9	15	106	3	5	3	1	641	800.18	1.40	360	0
inst_21	7	20	140	3	7	3	0	819	tl	1417.59	2959	122
	8	20	136	3	7	3	1	752	301.53	15.10	790	3
	9	20	152	4	7	4	0	958	tl	287.98	1941	24
inst_26	7	25	162	4	8	6	1	1169	tl	803.07	3106	52
	8	25	184	4	9	6	1	1017	tl	1.65	1040	4
	9	25	178	4	9	6	0	1173	tl	0.39	2320	0
inst_31	7	30	196	4	9	4	3	1097	tl	3.87	1630	0
	8	30	218	5	10	6	3	1069	tl	0.77	822	5
	9	30	206	5	10	7	1	1215	tl	1.24	1960	0
inst_36	7	35	272	6	13	13	2	1783	tl	0.00	1214	0
	8	35	242	5	11	8	1	1201	tl	129.83	886	15
	9	35	256	6	12	10	4	1577	tl	0.69	1196	2

TABLE A10 | Results of the BC-3IVF on newly generated instances.

Instances				Solution								
Name	**Class**	n	$\lvert R_j \rvert$	K_{min}	K	VH	CS	**Obj**	**Time$_T$**	**Time$_P$**	**Cut$_R$**	**Cut$_P$**
inst_11	7	10	68	2	4	2	0	534	1.17	0.97	14	0
	8	10	68	2	4	2	0	547	0.12	0.00	16	0
	9	10	60	2	3	2	0	516	0.18	0.00	20	0
inst_16	7	15	90	2	5	2	0	670	9.72	2.43	170	0
	8	15	122	3	6	3	0	758	26.57	0.18	120	2
	9	15	106	3	5	3	2	641	449.68	0.00	352	0
inst_21	7	20	140	3	7	3	0	815	tl	1046.63	2722	101
	8	20	136	3	7	3	0	752	391.57	78.27	1040	8
	9	20	152	4	7	4	2	973	tl	121.49	2216	24
inst_26	7	25	162	4	8	5	1	1030	tl	1713.88	1984	68
	8	25	184	4	9	6	4	1056	tl	4.04	2284	10
	9	25	178	4	9	5	1	1155	tl	125.87	2952	8
inst_31	7	30	196	4	9	6	3	1328	tl	7.75	2704	1
	8	30	218	5	10	6	3	1403	tl	164.53	2040	5
	9	30	206	5	10	9	7	2029	tl	0.07	2974	2
inst_36	7	35	272	6	13	8	6	1747	tl	23.73	1364	0
	8	35	242	5	11	10	3	1633	tl	0.17	1150	1
	9	35	256	6	12	12	8	2583	tl	0.15	1990	0