A variable neighborhood search for the green vehicle routing problem with two-dimensional loading constraints and split delivery

Kamyla Maria Ferreira*

Institute of Mathematics and Computer Sciences, University of São Paulo, ICMC-USP, São Carlos, Brazil.

Thiago Alves de Queiroz

Institute of Mathematics and Technology, Federal University of Catalão, IMTec-UFCAT, Catalão, Brazil.

Pedro Munari

Production Engineering Department, Federal University of São Carlos, DEP-UFSCar, São Carlos, Brazil.

Franklina Maria Bragion Toledo

Institute of Mathematics and Computer Sciences, University of São Paulo, ICMC-USP, São Carlos, Brazil.

Abstract

We address the Green Vehicle Routing Problem with Two-Dimensional Loading Constraints and Split Delivery (G2L-SDVRP), which extends the split delivery vehicle routing problem to include customer demands represented by two-dimensional, rectangular items. We aim to minimize carbon dioxide (CO_2) emissions instead of travel distance, a critical issue in contemporary logistics activities. The CO_2 emission rate is proportional to fuel consumption and measured in terms of the vehicle's total weight and traveled distance. We propose the first metaheuristic for the G2L-SDVRP, based on a variable neighborhood search approach that designs effective routes and guarantees the feasibility of loading constraints using various strategies, such as lower bound procedures, the open space heuristic, and a constraint programming model. We evaluate the performance of our approach through computational experiments using benchmark and newly created instances. The results indicate that the proposed approach is effective. It achieves improved solutions for 21 out of 60 instances in relatively short computing times when compared to existing methods for the G2L-SDVRP. Furthermore, our approach is competitive on benchmark instances of a related variant, namely the Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints, improving the best-known solutions for 50 out of 180 instances.

Keywords: Vehicle routing problem; Two-dimensional loading constraints; Split delivery; Greenhouse gas emissions; Variable neighborhood search.

*Corresponding author. Kamyla Maria Ferreira

Email addresses: kamylamaaria@gmail.com (Kamyla Maria Ferreira), taq@ufcat.edu.br (Thiago Alves de Queiroz), munari@dep.ufscar.br (Pedro Munari), fran@icmc.usp.br (Franklina Maria Bragion Toledo)

1. Introduction

Goods distribution is one of the most important logistics activities. For this reason, the Vehicle Routing Problem (VRP), which effectively models the main aspects of goods distribution, is one of the most well-known and extensively researched combinatorial optimization problems. Many studies have handled different variants of the VRP to satisfy the practical constraints that arise in real-life applications of distribution companies (Golden et al., 2008; Toth and Vigo, 2014). In this paper, we are interested in a variant known as the Green Vehicle Routing Problem with Two-Dimensional Loading Constraints and Split Delivery (G2L-SDVRP). It generalizes the VRP by incorporating practical restrictions on two-dimensional loading, split delivery, and environmental issues concerning carbon dioxide (CO_2) emissions. The G2L-SDVRP is a combination of the Capacitated VRP with Two-dimensional Loading Constraints (2L-CVRP) (Iori et al., 2007), the Split Delivery VRP (SDVRP) (Archetti et al., 2014; Munari and Savelsbergh, 2022), and the Pollution-Routing Problem (PRP) (Bektaş and Laporte, 2011). Since the G2L-SDVRP is a generalization of the VRP, it is also a challenging NP-hard problem.

In many situations, such as the distribution of household appliances, heavy machinery, and pallet cargoes, the loads are typically large, fragile, and cannot be stacked. As a consequence, the arrangement of items in the vehicles typically has a significant impact on the routes, especially if we consider unloading constraints (Iori et al., 2007). The unloading constraint, also known as the last-in-first-out (LIFO) constraint, imposes items of a customer on being unloaded from the vehicle without moving any items of other customers, motivated by the difficulty or even impossibility of moving items due to their weight and size (Nascimento et al., 2021). Therefore, by considering routing, packing, and unloading decisions simultaneously, we may prevent situations in which the designed routes cannot be associated with a feasible packing or schedule.

Moreover, each customer's demand can be higher than the vehicle capacity in real-world applications. Therefore, it is useful to resort to split delivery so that a customer can be visited by more than one vehicle when its demand exceeds the vehicle capacity. Previous studies indicate that allowing split delivery to customers, even if the demand is not higher than the vehicle capacity, may provide savings in the costs and number of used vehicles (Archetti et al., 2006). Our study is also motivated by the urgent need to reduce gas emissions and improve air quality in urban centers (Demir et al., 2014). Transportation activities influence the environment because it is a major consumer of petroleum and produces a significant amount of CO_2 emissions (Salimifard et al., 2012). Therefore, it is necessary to consider the environmental impact of freight transportation while planning the routing schedule.

The G2L-SDVRP incorporates all the practical motivations mentioned above. It consists of determining vehicle routes that minimize the amount of CO_2 emitted while satisfying all customer demands. These demands correspond to two-dimensional rectangular items that must be loaded onto the vehicle's rectangular base without overlapping and respecting the base dimensions, besides satisfying unloading constraints. If the splitting is beneficial, customers can be served by one or more vehicles, where each vehicle transports a fraction of the demand (i.e., a part of the customers' items).

1.1. Related literature

To the best of our knowledge, the only available solution methodology for the G2L-SDVRP is the exact approach developed by Ferreira et al. (2021). The authors proposed a tailored branch-and-cut method with specific procedures to handle the packing subproblem. Due to the difficulty of solving this subproblem, the authors used different strategies such as heuristics, lower bound procedures, and a constraint programming model. Additionally, a hash table to save routes already checked was used to reduce the computational effort, while a pattern (grid) of points was used to reduce the number of available points to pack items. The method solved instances with up to 35 customers and 114 items, where only 23 out of 60 instances were optimally solved. These authors compared the solutions of the G2L-SDVRP with those of three other problems, namely the 2L-CVRP, the Vehicle Routing Problem with Two-Dimensional Loading Constraints and Split Delivery (2L-SDVRP), and the Green Vehicle Routing Problem with Two-Dimensional Loading Constraints (G2L-CVRP). The results indicated that solving the G2L-SDVRP is the best choice overall for practical purposes, with an average percentage difference of 1.69%, 5.82%, and 3.65% in comparison to the G2L-CVRP, 2L-SDVRP, and 2L-CVRP, respectively. Moreover, incorporating environmental issues reduces emissions, while the possibility of split delivery makes it possible to minimize emissions even further.

Other studies have addressed the combination of the 2L-CVRP with split deliveries (Annouch et al., 2016; Ji et al., 2021) and the SDVRP with environmental considerations (Vornhusen and Kopfer, 2015; Matos et al., 2018). Annouch et al. (2016) proposed an exact branch-and-cut approach to solve the 2L-CVRP with split delivery and additional constraints motivated by the distribution of liquid petroleum gas. Ji et al. (2021) addressed another variant of the 2L-CVRP with split delivery, in which items can be rotated by 90° and relocated during unloading operations at customers. The authors proposed an enhanced neighborhood search algorithm combined with the maximum-space-utilization heuristic to solve the problem. Vornhusen and Kopfer (2015) proposed an exact method based on branch-and-cut for the SDVRP with time windows, a heterogeneous fleet, and CO_2 emissions. The problem aims to reduce CO_2 emissions, estimated according to the total weight of the vehicles in each arc. Matos et al. (2018) developed a hybrid algorithm that combines an iterated local search, random variable neighborhood descent procedure, and a set covering model for the green vehicle routing and scheduling problem, considering the minimization of CO_2 emission. The problem involves a heterogeneous fleet of vehicles that can perform split deliveries to customers and assumes time-varying network traffic congestion. The authors measured the CO_2 emission by observing vehicle speed, weight, and traveled distance.

It is worth mentioning that there are different exact and heuristic methods for standalone variants of the 2L-CVRP (Iori et al., 2007; Zachariadis et al., 2013; Wei et al., 2015; Côté et al., 2017; Wei et al., 2018; Silva et al., 2022; Zhang et al., 2022), SDVRP (Dror et al., 1994; Archetti et al., 2014; Silva et al., 2015; Shi et al., 2018; Munari and Savelsbergh, 2020, 2022; Balster et al., 2023), and PRP (Bektaş and Laporte, 2011; Zhang et al., 2014; Ehmke et al., 2016; Dabia et al., 2016; Dewi and Utama, 2021). Another closely related problem is the split delivery vehicle routing problem with three-dimensional loading constraints (3L-SDVRP). It considers the packing of three-dimensional items. There is a very limited number of studies on this problem, and they involve heuristics based on one-stage local search (Ceschia et al., 2013), data-driven three-layer search (Li et al., 2018), tabu search (Yi and Bortfeldt, 2018), local search (Bortfeldt and Yi, 2020), and column generation (Rajaei et al., 2022). We refer to Pollaris et al. (2015); Archetti and Speranza (2012); Lin et al. (2014) and Krebs and Ehmke (2023) for more details and overviews.

1.2. Our contributions

The literature review on the G2L-SDVRP and related problems shows limited studies on VRP variants, including split delivery and CO_2 emissions. Notably, these studies clearly indicate the importance of including such features in solution approaches, as they improve the quality of the solutions regarding practical aspects (Ferreira et al., 2021; Ji et al., 2021; Bortfeldt and Yi, 2020). For example, Ferreira et al. (2021) show that the gains from considering split delivery and CO_2 emissions are superior to 1%. However, since they managed to solve only small instances using their approach, there is a lack of effective solution approaches for medium and large-sized instances of this problem. Hence, we close this gap by proposing a metaheuristic to effectively solve large instances, i.e., instances with more customers.

We develop the first metaheuristic for the G2L-SDVRP, which is a Variable Neighborhood Search (VNS), motivated by the outstanding performance of this approach on related VRP variants (Hemmelmayr et al., 2009; Imran et al., 2009; Wei et al., 2015; Xiao and Konak, 2016; Ferreira et al., 2018; Sadati and Çatay, 2021). We are not aware of any other metaheuristic approach proposed for this problem. Our implementation relies on five neighborhood operators, a local search based on the random variable neighborhood descent, a set partitioning model in the intensification phase, a diversification procedure to escape from local optima, and a procedure with different strategies to quickly check the feasibility of packings. Our method searches only in the feasible solution space.

The main difference between our approach and that of Ferreira et al. (2021) lies in the methodology used to address the problem. We develop a metaheuristic based on VNS, while Ferreira et al. (2021) introduce an exact algorithm with a branch-and-cut technique. Both approaches rely on similar packing procedures; however, in contrast to Ferreira et al. (2021), we consider only those procedures with low computational effort. Moreover, we incorporate several enhancements, including a technique for adjusting the dimensions of items when there is unused space in the vehicle base, a more sophisticated heuristic for packing items, and a pattern of points that considers the unloading requirements. It is important to note that these adjustments are necessary due to the divergent nature of the two approaches, each requiring components better suited to its respective purpose.

In summary, the main contributions are: *(i)* the introduction of the first metaheuristic for the G2L-SDVRP; *(ii)* an ad hoc solution representation scheme for the G2L-SDVRP; *(iii)* the proposal of specific neighborhood operations to generate solutions with split delivery; *(iv)* a procedure to reduce the feasible positions of items in the solution vector of the packing problem, which is based on the unloading constraint; and *(v)* new bounds and improved solutions for benchmark and new instances of the G2L-SDVRP and 2L-CVRP.

Computational experiments with benchmark instances indicate that the proposed approach can provide high-quality solutions in relatively short computing times. More precisely, it obtains the same best-known solutions reported in the literature for 32 (out of 60) instances and improves the solutions of the other 21 instances, with an average and maximum improvement in the objective value of 0.38% and 9.38%, respectively. Furthermore, when applied to solve benchmark instances of the 2L-CVRP, the results show that our method is competitive with state-of-the-art approaches. It finds the best-known solution for 97 (out of 180) instances and improves the records for 50 other instances of the 2L-CVRP.

The remainder of this paper is organized as follows. Section 2 describes the G2L-SDVRP. Section 3 presents the proposed VNS metaheuristic. Section 4 introduces the procedure for checking packing feasibility. Section 5 discusses the computational experiments. Finally, concluding remarks and suggestions for future works are given in Section 6.

2. Problem description

The G2L-SDVRP can be defined on a complete directed graph $G = (N, A)$, where $N = \{0, 1, \ldots, n\}$ is the set of nodes and $A = \{(i, j) \,|\, i, j \in N, i \neq j\}$ is the set of arcs. Node 0 represents the central depot, and the remaining nodes denote the customers. Each arc $(i, j) \in A$ is associated with a travel distance D_{ij}, which, for the sake of simplicity, we assume is proportional to the travel cost. There is a set $K = \{1, \ldots, K_{max}\}$ of K_{max} identical vehicles available at the depot. Each vehicle $k \in K$ has weight capacity Q and a rectangular loading surface/base of width W and length L, whose total area is $A_T = W \times L$.

Each customer j demands a set R_j of rectangular items with total weight $P_j = \sum_{r=1}^{|R_j|} p_{jr}$ and total area $A_j = \sum_{r=1}^{|R_j|} a_{jr}$. Each item $r \in R_j$ has width w_{jr}, length l_{jr}, weight p_{jr}, and area $a_{jr} = w_{jr} \times l_{jr}$. Each rectangular item is described by a pair (j, r), where r is the item index. A feasible solution for the problem satisfies the following constraints:

- each vehicle, if used, starts and ends its route at the depot;
- the number of routes is less than or equal to the number of vehicles;
- the demand assigned to each route does not exceed the vehicle capacity in terms of weight and area;
- each customer is served by at least one vehicle, and her total demand is satisfied;
- each vehicle visits a customer only once;
- each item has a fixed orientation and cannot be rotated during the packing;
- each item is loaded with its edges parallel to those of the vehicle base;
- items do not overlap when packed in the same vehicle;
- items are not rearranged during the unloading operation at customers.

The objective function aims to minimize the amount of CO_2 emitted by executing the planned routes. The CO_2 emission is calculated based on the number of liters of fuel consumed, measured in terms of the traveled distance and the weight of the fully loaded and empty vehicle (Xiao et al., 2012). Therefore, the amount of CO_2 emission in each arc $(i, j) \in A$ is given by $ER_{CO_2}(\rho_0 + \frac{\rho_f - \rho_0}{Q} f_{ij})D_{ij}$, where ER_{CO_2} is the CO_2 emission rate per liter of fuel consumed; ρ_0 and ρ_f are constants that represent the fuel consumption rate when the vehicle is empty and fully loaded, respectively; and, f_{ij} is the transported load in the arc (i, j). We set the values of ρ_0 and ρ_f to 1 and 3, respectively, as in the previous study of Ferreira et al. (2021). We refer to the same paper for a complete mathematical formulation of the G2L-SDVRP.

3. The variable neighborhood search metaheuristic for the G2L-SDVRP

The proposed approach consists of a multi-start metaheuristic mainly based on VNS (Mladenović and Hansen, 1997). The VNS metaheuristic explores the solution space by systematically changing neighborhoods when an improvement move is not found. In general, the steps of our VNS metaheuristic can be summarized as (*i*) generate an initial solution; (*ii*) shake the solution by applying neighborhood structures; (*iii*) apply the local search; (*iv*) perform intensification on the solution; and, (*v*) diversify the best solution. Furthermore, whenever a new route is found during any step (*i*)-(*v*), the approach verifies the feasibility of the packing involving the items of customers in the route. If the packing is infeasible, the route is discarded since only feasible solutions are accepted.

The pseudo-code of the developed VNS is presented in Algorithm 1. The algorithm has two input parameters: NN is the maximum number of consecutive iterations allowed without improving the best solution, T_{max} is the time limit, and K_{max} is the total number of vehicles at the depot. The best solution is represented by X^*. In Section 3.1, we describe how a solution is represented. Section 3.2 describes the procedure that constructs the initial solution X. The routes of the initial solution are stored in a pool $P_{partition}$. After that, solution X is submitted to the local search procedure described in Section 3.4. In the loop of lines 12-18, a neighbor solution X' is obtained using one of the five neighborhood structures described in Section 3.3. If an improved solution is found, the sequence with the neighborhood structures V is shuffled randomly, as we adopt a random ordering of neighborhoods. After K_{max} iterations, if the best solution X^* is improved, the counter nn is reinitialized. Thus, the set partitioning problem is solved as described in Section 3.5, and if the solution X is better than X^*, we update X^* accordingly. Otherwise, the diversification procedure is applied, following the procedure given in Section 3.6. The algorithm ends when the time limit T_{max} is reached or the best solution X^* is not improved after NN consecutive iterations. After all, the best solution X^* is returned.

Algorithm 1: VNS metaheuristic for the G2L-SDVRP.

1 **Input:** NN, T_{max}, K_{max};
2 **Output:** Best solution found;
3 Construct the initial solution X;
4 $P_{partition} \leftarrow$ Add the routes of X into the route pool;
5 $X \leftarrow$ Apply the local search on X;
6 $X^* \leftarrow X$; $nn \leftarrow 0$;
7 Define the set of neighborhood structures $V = \{V_1, V_2, V_3, V_4, V_5\}$;
8 **while** $time < T_{max}$ ***and*** $nn < NN$ **do**
9     $nn \leftarrow nn + 1$;
10   **for** $k \leftarrow 1$ ***to*** K_{max} **do**
11         $v \leftarrow 1$;
12   **while** $v \leq 5$ **do**
13   Generate a random neighbor $X^{'}$ of X using V_v;
14             $X^{''} \leftarrow$ Apply the local search on $X^{'}$;
15   **if** $X^{''}$ *is better than* X **then**
16                 $X \leftarrow X^{''}$; $v \leftarrow 0$;
17   Shuffle the order of the neighborhood structures V;
18             $v \leftarrow v + 1$;
19   **if** X *is better than* X^* **then** $X^* \leftarrow X$; $nn \leftarrow 0$;
20     $X \leftarrow$ Solve the set partitioning problem on X^*;
21   **if** X *is better than* X^* **then** $X^* \leftarrow X$; $nn \leftarrow 0$;
22   **else** $X \leftarrow$ Apply the diversification procedure on X^*;
23 **return** X^*;

3.1. *Solution representation*

With the possibility of splitting deliveries in the G2L-SDVRP, it is necessary to determine and store which customer items will be in each vehicle. Hence, solution encoding is very important to make an effective method. In our implementation, a solution X is represented as a set of sequences r_k, for $k = 1, \ldots, , K_{max}$. For each vehicle k, r_k represents the sequence of customers served by vehicle k in the order they will be visited. Additionally, for each customer $i = 1, \ldots, n$, we create a sequence S_i containing the vehicle index that will serve each item of customer i.

Figure 1 illustrates the representation of a solution for a given G2L-SDVRP instance. There is a central depot (node 0), 10 customers (nodes 1 to 10), and 3 vehicles (i.e., $K_{max} = 3$). The first five customers (1 to 5) require two items each, and the last five (6 to 10) require three. Customers 9, 10, and 1 are served by route r_1; customers 6, 3, 8, and 5 are served by r_2; and customers 6, 2, 4, and 7 are served by r_3. Customer 6 is served by two different routes (r_2 and r_3), where items 1 and 3 are delivered by vehicle 2, and item 2 is delivered by vehicle 3.

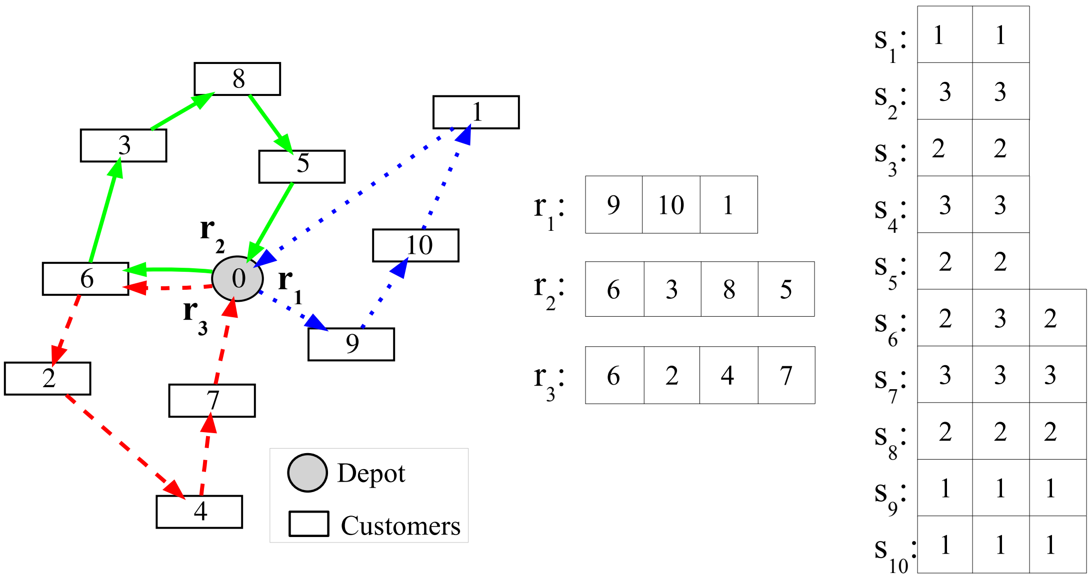


Figure 1: An example of a solution and its representation for a given G2L-SDVRP instance.

3.2. *Initial solution*

The initial solution is constructed using the two-phase procedure of Wei et al. (2018). In the first phase, routes are generated by the savings algorithm (Clarke and Wright, 1964). This algorithm starts with single-customer routes that have no split deliveries, i.e., for each customer $i = 1, \ldots, n$, it creates the route $(0 - i - 0)$. Next, the savings $(D_{i0} + D_{0j} - D_{ij})$ are calculated and sorted in descending order. In each step, two routes are merged according to the largest savings. For this, the arc (i, j), from the top of the list of savings, is considered. If customers i and j can be merged, and the vehicle capacity and loading constraints are respected, the arc (i, j) is added, and then the arcs $(i, 0)$ and $(0, j)$ are removed. Notice that all savings are calculated without considering the CO_2 emission; only the route costs are used. Given that the calculation of the CO_2 emission is based on the weight transported between two nodes, it would be necessary to recalculate the savings after merging any two routes, requiring extra computing time. In preliminary computational experiments using this recalculation, the procedure required up to 1200 seconds to obtain an initial solution for some instances. For this reason, we decided to ignore CO_2 emission in the savings calculation. For the same reason, split deliveries were not considered in this procedure either. We instead rely on specific local search operators to generate split deliveries, as this strategy has proven to be more efficient in the computational tests.

The savings algorithm ends when no further route merge is possible/feasible. If the number of routes is less than or equal to the number of available vehicles, the procedure returns the constructed routes as a feasible solution; otherwise, it starts the second phase. In each iteration of this phase, the route with the lowest utilization rate of the vehicle base is eliminated, and its customers are added to a pool. These customers are sorted by decreasing area and reinserted into the solution using the cheapest insertion algorithm. In other words, each customer is inserted into the position and route with the lowest incremental cost, respecting the problem constraints. One route is randomly selected when a customer cannot be inserted into any route because of not respecting the vehicle capacity and loading constraints. Customers in this route are successively removed and added to the pool until the given customer is inserted into this route. Thus, the reinsertion procedure of the customers in the pool is restarted.

3.3. Neighborhood structures

We use five neighborhood structures in our implementation (line 7 of Algorithm 1), which are based on the literature of (meta)heuristics for solving the 2L-CVRP (Zachariadis et al., 2013; Wei et al., 2015, 2018; Ji et al., 2021), SDVRP (Silva et al., 2015; Matos et al., 2018) and other VRP variants. They are:

- Customer relocation: a customer is relocated to another position;
- Route exchange: the positions of two customers are exchanged;
- Route interchange: two positions i and j are selected. If they are on the same route (intra-route), the segment of customers between i and j (including them) is considered in reverse order. When i and j belong to different routes (inter-route), the first part of the route that is before i is connected with the second part of the route that is after j, and the second part of the route after that is after i is connected with the first part of the route that is before j;
- Block exchange: the positions of two segments are exchanged;
- Block relocation: a segment of customers is relocated to another position.

Each neighborhood structure can perform operations in a single route (intra-route) and two routes (inter-route). All position and route choices are random in the shaking step. The sequence/segment size is limited to four positions in neighborhoods *block exchange* and *block relocation*. It is worth mentioning that other values were tested in preliminary experiments, but the best results have been obtained using the sequence limited to four positions.

3.4. Local search

The local search relies on the *randomized variable neighborhood descent* (RVND) algorithm (Subramanian et al., 2010). An important issue related to the variable neighborhood descent is the order in which the local search operators are applied. To overcome this difficulty, we randomly generated the order in which the local search operators are considered. By incorporating randomness into the (deterministic) variable neighborhood descent algorithm, this strategy avoids an extra parameter to define the neighborhood order, which needs to be calibrated. A similar strategy was used by, e.g., Subramanian and Battarra (2013); Penna et al. (2013); Silva et al. (2015); Wei et al. (2015); Matos et al. (2018). Our RVND adopts the first improvement strategy, i.e., the local search tries all possible movements of an operator until reaching the first one that results in a solution better than the current. In addition, if some local search operator improves the current best solution, the improved solution is added to the route pool $P_{partition}$. Algorithm 2 describes our RVND.

Algorithm 2: Local search based on the Random Variable Neighborhood Descent.

Input: X', T_{max}, p_{max};
Output: Best solution found;
$X'' \leftarrow X'$;
$p \leftarrow 1$;
$P \leftarrow \{1, \ldots, p_{max}\}$;
Shuffle the order to apply the local search operators P;
while $p \leq p_{max}$ ***and*** *time* $< T_{max}$ **do**
  $X'' \leftarrow$ Apply the local search P_p on X';
  if X'' *is better than* X' **then**
    $X' \leftarrow X''$; $p \leftarrow 1$;
    Shuffle the order to apply the local search operators P;
    Add the routes of X'' to the pool $P_{partition}$;
  $p \leftarrow p + 1$;
return X';

We develop ten local search neighborhood structures. Three of them consider intra-route operations, while seven are related to inter-route operations. Concerning the ones based on inter-route operations, two are specific to handling split deliveries. Six of the ten neighborhood structures consider both intra- and inter-route operations. They are customer relocation (intra-route), customer relocation (inter-route), route exchange (intra-route), route exchange (inter-route), route interchange (intra-route), and route interchange (inter-route). These local search neighborhoods are based on the neighborhood structures in Section 3.3. The others are neighborhood structures based on inter-route operations, i.e.:

- Exchange(2, 1): exchange the positions of two adjacent customers with a customer of another route;
- Relocation(2, 0): two adjacent customers are removed from one route and inserted in another route;
- Split-delivery relocation: given two positions i and j from different routes, one customer item at position i is removed and inserted before position j. Figure 2 shows an example in which customer 5 is served with split delivery. Only item 2 of customer 5 can be moved to route r_1 without violating the vehicle capacity and loading constraints.
- Split-delivery exchange: given two positions i and j from different routes, the customer at position i is inserted before the customer at position j and one of the items of the customer

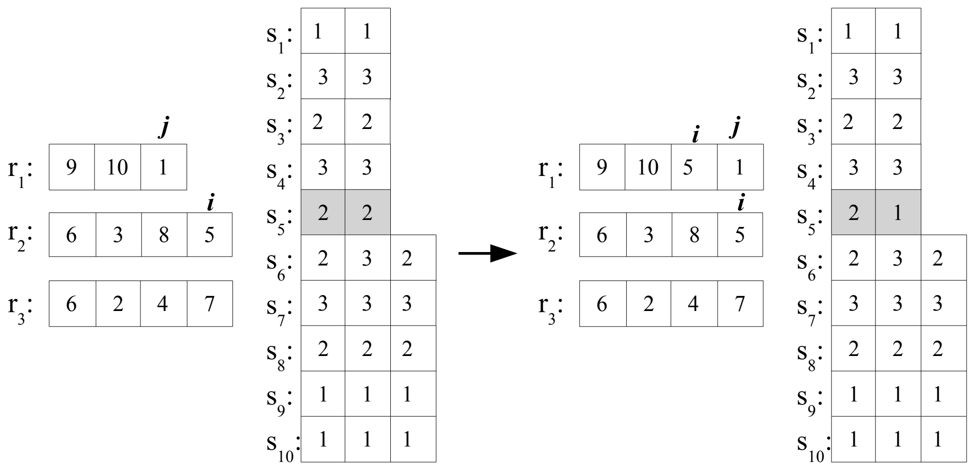


Figure 2: Example of a split-delivery relocation operation.

at position j is inserted before position i. Figure 3 shows an example in which customer 5 is served by routes r_1 and r_2, and customer 1 is moved to route r_2.

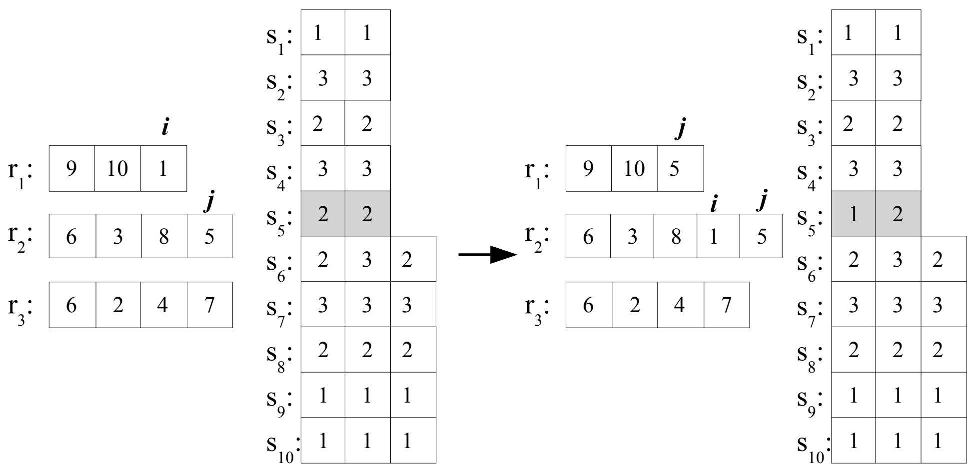


Figure 3: Example of a split-delivery exchange operation.

We tested other neighborhood structures, such as block exchange and block relocation, but since they were time-consuming without consistently improving the final solution, we decided not to include them in our method. Moreover, aiming to reduce the computing times, we used a hash structure to keep track of the routes that cannot be further improved by applying the local search neighborhood structures. Since routes obtained by removing one or more customers always result in feasible packing, there is no need to solve their packing subproblem. This observation is considered in customer relocation (intra- and inter-route), relocation(2,0), and split-delivery relocation neighborhood structures. A hash structure is created for each local search neighborhood. In intra-route operations, only information from one route is stored, while in inter-route operations, information from both routes is kept. For each route, the key is a string with the customer sequence, cost, demand, and total area, such that the customers are separated by "|" and the cost, demand,

and area are separated by "-". Additionally, in the inter-route operations, the routes are separated by "+". Aiming to reduce the computational effort, we evaluate the solution cost before solving the packing subproblem. This means there is no need to check the packing feasibility of routes costing more than those in the current solution.

3.5. Intensification procedure based on the set partitioning problem

We solve exactly a variant of the set partitioning problem (SPP) in the intensification procedure of the proposed VNS. Let SR be the set of all feasible routes known for an instance. They are stored in the pool $P_{partition}$. We define $SR_i \subseteq SR$ as the subset of all routes that contain customer $i \in N \setminus \{0\}$; C_k as the cost of route $k \in SR$; and τ_{ik} as the number of items of customer i served by route k. The SPP formulation is composed of the objective function (1) and constraints (2) to (5). The decision variable ϕ_k is equal to 1 if route k is chosen; and 0 otherwise.

$$\min \sum_{k \in SR} C_k \, \phi_k, \tag{1}$$

$$\text{s.t.} \sum_{k \in SR_i} \tau_{ik} \phi_k = |R_i|, \quad \forall i \in V \setminus \{0\}, \tag{2}$$

$$\phi_s = \phi_k, \quad \forall k, \, s \in SR: \; k \text{ and } s \text{ have at least one split delivery in common}, \tag{3}$$

$$\sum_{k \in SR} \phi_k \leq K_{max}, \tag{4}$$

$$\phi_k \in \{0, \, 1\}, \quad \forall k \in SR. \tag{5}$$

The objective function (1) aims to minimize the total cost of the chosen routes. Constraints (2) ensure that all items R_i of customer i are delivered. Constraints (3) guarantee that routes having at least one split delivery customer in common are in the solution, i.e., if a route with a partial delivery to a customer is selected, then any other route serving this customer with partial delivery must also be selected. These constraints were adapted from Matos et al. (2018). Constraint (4) ensures that the number of routes does not exceed K_{max}. Constraints (5) define the domain of the variables.

Given that the number of feasible routes is exponential in the instance size, we limit the number of routes in SR to be Nr, a parameter we set in advance. When we find Nr routes, we solve the formulation and delete all routes except those in the incumbent solutions obtained from the SPP formulation. Hence, the best solutions are preserved. After preliminary tests, we set Nr to 10000. Moreover, we rely on hashing strategies to avoid duplicate routes, and the best solution is provided as the initial solution when solving the SPP formulation.

3.6. Diversification procedure

We propose a diversification procedure that significantly changes the best solution and avoids the VNS becoming stuck in local optima solutions. We consider procedure based on the ruin-reconstruct mechanism of Wei et al. (2015). In the ruin process, Nc customers are randomly removed from the solution and inserted into a pool. If a customer with split delivery is selected, it is removed from all routes that visit her. Next, the reconstruction step generates the solution as described in Section 3.2. In accordance with Wei et al. (2015), parameter Nc is defined as $\min\{0.5 \times n, 0.1 \times n + nn\}$, where n is the total number of customers and nn is the number of VNS consecutive iterations without improving the best solution.

4. A heuristic approach for the two-dimensional loading subproblem

The procedure to check the feasibility of a route due to loading constraints is frequently invoked by our VNS method. Thus, it is essential to have a fast and effective approach. We use six procedures to quickly determine the feasibility of a route, including lower bounds, heuristics, solving a mathematical model, updating items' dimensions, and using a hash structure. The hash structure keeps track of the routes already checked due to the packing to reduce computational effort. In this structure, each route is associated with a key given by the sequence of customers and items. In addition, some procedures do not consider the unloading requirements. Thus, if a route is infeasible, it implies that any sequence permutation involving those customers is also infeasible. Therefore, in our hash structure, each key is associated with one of the following three status values: 1, a route with a feasible packing (Procedures 5 and 6); -1, a route with an infeasible packing (Procedures 4 and 6); and -2, a route with an infeasible packing, regardless of the sequence in which customers are to be visited (Procedures 1, 2 and 3).

The six procedures are called sequentially until the packing is proven feasible or infeasible. Initially, we check whether the route is already in the pool of hashed routes. If this is true, we return its status; otherwise, we apply, next, the procedure proposed by Boschetti et al. (2002) to update items' dimensions. This procedure preserves optimality and increases the item width and length in accordance with the unused area of the vehicle base. The width and length of each item $i \in M$, given M as the set of all items of the customers on a route, are updated using (6) and (7), respectively. Consequently, the new dimension of item $i \in M$ in terms of width becomes $w_i + (W - w_i^*)$, and in terms of length it becomes $l_i + (L - l_i^*)$. The problem in (6) and (7) consists of a one-dimensional knapsack problem, which we solve using the dynamic programming algorithm presented by Martello and Toth (1990). More details about this and related procedures can be found in Almeida Cunha et al. (2020).

$$w_i^* = \max\left\{ w = \sum_{j \in I \setminus \{i\}} \varepsilon_j w_k + w_i \mid w \leq W,\ \varepsilon_j \in \{0, 1\},\ j \in I \setminus \{i\} \right\}, \tag{6}$$

$$l_i^* = \max\left\{ l = \sum_{j \in I \setminus \{i\}} \varepsilon_j l_k + l_i \mid l \leq L,\ \varepsilon_j \in \{0, 1\},\ j \in I \setminus \{i\} \right\}. \tag{7}$$

After updating the items' dimensions, Procedure 1 is applied. In the next step, we calculate the total area of items and then determine in which order to apply Procedures 2-6. If the total area is less than 80% of the vehicle area, there is a high chance the route will be feasible for packing, so the following order is considered: Procedures 5, 4, 3, 2, and 6; otherwise, we consider Procedures 2, 3, 4, 5, and 6, in this order. In addition, in Procedures 1 to 4, if the lower bound value is larger than the length of the vehicle base, the route packing is infeasible. Algorithm 3 describes the procedure for checking whether a packing is feasible considering the set of M items.

Procedure 1: a lower bound of the minimum length required to pack all items is obtained from dividing the sum of the areas of the items in M by the width of the vehicle base.

Procedure 2: a lower bound on the required length of the loading area is estimated by the alternate constructive procedure of Alvarez-Valdés et al. (2009). This procedure changes items' dimensions. If the modified items do not fit in the vehicle base, then the original instance has no feasible packing.

Procedure 3: a lower bound on the minimum length required to pack all items in M is estimated by dual feasible functions. We consider only the first three dual feasible functions described in (Boschetti et al., 2002) since the fourth may require a high computational effort. Côté et al. (2017) adopted a similar strategy.

Procedure 4: a lower bound from Côté et al. (2014) on the minimum length of the vehicle base is calculated considering unloading requirements. The idea is to constrain positions in which items can be on the length of the vehicle base.

Procedure 5: a Randomized Local Search (RLS) metaheuristic combined with the Open Space technique (Wei et al., 2018). This method is called RLS+OP and is detailed in Section 4.1.

Procedure 6: a constraint programming (CP) model based on solving the non-preemptive cumulative-scheduling problem (Clautiaux et al., 2008). We reduce the domain of the decision variables by considering the grid of normal patterns (Herz, 1972). If CP provides no feasible solution, then the route is infeasible. We apply CP to check only the packing of routes obtained using the local search procedures and only if the percentage difference between the current solution and the modified solution is greater than 0.5%.

Algorithm 3: Solving the packing subproblem.

```
Input: S, M;
Output: Whether route S has a feasible packing;
feas ← False;
if S is not in the hash pool then
    Update the dimensions of items in M;
    feas ← Apply Procedure 1;
    if feas is False then
        if total area of items in S is less than 80% then LP ← Consider Procedures {2, 3, 4, 5, 6}
         ;
        else LP ← Consider Procedures {5, 3, 4, 2, 6} ;
        foreach p ∈ LP do
            feas ← Apply Procedure p;
            if feas is True and p is equal to Procedure 5 then Break the loop ;
            else if feas is False and p is different of Procedure 5 then Break the loop ;
        Add S into the hash pool with the status in feas;
else feas ← Status of the packing for the route S ;
return feas;
```

4.1. RLS+OP metaheuristic

We propose the RLS+OP algorithm inspired by the sequence-based random local search method and the open space heuristic, both from Wei et al. (2018). In our algorithm, RLS generates the sequence/order in which items will be packed in the vehicle base. The open space technique is applied to pack the items following the given sequence. For the sake of simplicity, we consider M

as the set of all items in a route and σ as the order in which these items are packed in the vehicle assigned to this route.

A vector of ordered items represents the solution in the RLS+OP. Because the items' order greatly influences the algorithm performance, we develop a procedure to reduce the possible positions in which items can be allocated in the vector solution. Based on the unloading requirements, we estimate the indices of the minimum and maximum positions each item can be in the sequence/vector. First, we check whether two items with different visit orders cannot be packed side by side in the vehicle base; because of the unloading constraint, items with orders greater than i (i.e., that must be visited after i) cannot obstruct the unloading path of i. Figure 4 illustrates this scenario, where the hatched region marks the area where no item j, with $\sigma_j > \sigma_i$, can be since it blocks item j during the unloading operation (note that rehandling items is not allowed). The pseudo-code of the procedure to estimate the minimum (Pos_{min}) and maximum (Pos_{max}) positions is shown in Algorithm 4.

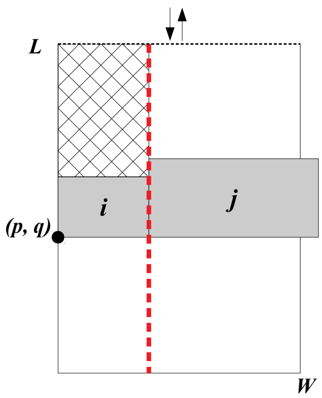


Figure 4: Illustrating items' (minimum and maximum) position in the solution vector.

First, the minimum position is determined by sorting items (σ) in increasing delivery orders. In the case of a tie, the item with the largest width is considered first. Next, for each pair of positions i and j, we check whether the item in position M_i is delivered before M_j and the sum of their widths does not exceed the width of the vehicle base. If both conditions are true, the items in positions M_i and M_j cannot be packed side by side. Therefore, the item in position M_i must be packed below the item in position M_j and, consequently, it must be in the solution vector after the position j. This combination of positions i and j, to define the minimum position of the item in M_i, is applied until finding an item in M_j whose order is equal to M_i. On reaching this criterion (line 9 of Algorithm 4), if the width of M_i is equal to the width of the vehicle base, the position j is a limit for M_i.

In the case of the maximum position in Algorithm 4, items are sorted in decreasing delivery orders, but the tie-breaking criterion is by the item with the smallest width. For each position i, it is verified whether the item in M_i has a width equal to the vehicle base. If true, the item in M_i has i as the maximum position in the solution vector. Otherwise, for each position i and j, it is checked whether the order of the item in M_j is smaller than the one in M_i and the sum of the items' widths is greater than the width of the vehicle base. If these conditions are true, the maximum position of the item in M_i is $j - 1$.

For each route, the first step in the RLS+OP in Algorithm 5 is to compute the minimum and maximum positions each item in M can be in the solution vector. We consider $Iter_{max}$ iterations

Algorithm 4: Procedure to calculate items' minimum and maximum positions in the solution vector.

Input: M, set of items with dimensions (w_i, l_i) for $i \in M$; σ, order in which items are packed; W, width of the vehicle base;
Output: Minimum and maximum positions that each item can have in the solution vector;
$m \leftarrow$ number of items in set M;
$M' \leftarrow M$;
$\sigma'_i \leftarrow \sigma_i$, $w'_i \leftarrow w_i$, for $i \leftarrow 1, \ldots, m$;
$Pos_{min}(i) \leftarrow 0$, for $i \leftarrow 1, \ldots, m$; `// Minimum position`
$M' \leftarrow$ Sort items in M', as well as w' and σ', in decreasing order of visit, breaking ties by choosing the item with the largest width first;
for $i \leftarrow 1$ ***to*** m **do**
    for $j \leftarrow 1$ ***to*** m **do**
        if $\sigma'_j > \sigma'_i$ ***and*** $w'_i + w'_j > W$ **then** $Pos_{min}(M'_i) \leftarrow j + 1$;
        else if $\sigma_j = \sigma_i$ **then**
            if $w'_i = W$ **then** $Pos_{min}(M'_i) \leftarrow j$;
            Break the loop;
$Pos_{max}(i) \leftarrow m$, for $i = 1, \ldots, m$; `// Maximum position`
$M' \leftarrow$ Sort items in M', as well as w' and σ', in decreasing order of visit, breaking ties by choosing the item with the smallest width first;
for $i \leftarrow 1$ ***to*** m **do**
    if $w'_i = W$ **then** $Pos_{max}(M'_i) \leftarrow i$;
    else
        for $j \leftarrow i + 1$ ***to*** m **do**
            if $\sigma'_j < \sigma'_i$ ***and*** $w'_i + w'_j > W$ **then**
                $Pos_{max}(M'_i) \leftarrow j - 1$;
                Break the loop;
return Pos_{min} *and* Pos_{max};

of the RLS algorithm to check the packing feasibility of a route. Besides that, we obtain an initial solution by using three sorting rules, which are: decreasing order by area (Od_1), decreasing order by length (Od_2), and decreasing order by width (Od_3). Given a sequence of items, the open space heuristic performs the packing, which returns the total area packed and the position of the last item packed in the vehicle base. If the packed area ($packed_{area}$) is equal to the total area of items ($total_{area}$), we have a feasible packing for this route. In the loop of lines 14–25, two items have their positions swapped, and the new sequence is submitted to the open space heuristic. If the packed area of the solution is equal to the total area, the procedure ends with the status *True* (i.e., a feasible solution is found). After all, the algorithm returns the status *False*, which, in this case, means an undefined solution.

The open space heuristic packs item by item, following the sequence generated by the RLS. In this heuristic, a packing pattern is represented by the packed region (i.e., the area occupied by the packed items) and the unpacked region (i.e., the union of all the free spaces, rectangular areas not occupied by an item). An open space is a free space with one side that coincides with the vehicle's rear door. The heuristic consists of updating the open spaces, which are candidate positions for positioning items, whenever a new item is packed. An item is packed in the open space with the

Algorithm 5: RLS+OP metaheuristic of the Procedure 5.

Input: M, sequence of items i with dimensions (w_i, l_i);
Output: Whether a feasible packing for M exists.;
Construct an initial solution X;
$Pos_{min}, Pos_{max} \leftarrow$ Calculate the valid positions of items by Algorithm 4;
$m \leftarrow$ number of items in M;
$total_{area} \leftarrow$ sum of the area of all items in M;
$Iter_{max} \leftarrow \max\{m, \lceil 100 \times (1 - (\frac{total_{area}}{A_t}))\rceil\}$;
for $i \leftarrow 1$ ***to*** $Iter_{max}$ **do**
    for $t \leftarrow 1$ ***to*** 3 **do**
        Sort the items in M using the sorting rule Od_t;
        $packed_{area} \leftarrow$ Apply the open space heuristic given M;
        $pos \leftarrow$ position of the last item packed by the open space heuristic;
        if $packed_{area} = total_{area}$ **then return** *True* ;
        for $j \leftarrow 1$ ***to*** m **do**
            $M' \leftarrow$ randomly swap the position of two items in M;
            $packed'_{area} \leftarrow$ Apply the open space heuristic given M';
            if $packed'_{area} > packed_{area}$ **then**
                $j \leftarrow 1$;
                $M \leftarrow M'$;
                $packed_{area} \leftarrow packed'_{area}$;
                $pos \leftarrow$ position of the last item packed by the open space heuristic;
                if $packed_{area} = total_{area}$ **then return** *True* ;
            else if $packed'_{area} = packed_{area}$ **then**
                $M \leftarrow M'$;
                $pos \leftarrow$ position of the last item packed by the open space heuristic;
return *False*

smallest y-coordinate that respects the unloading constraint. The algorithm aims to pack as many items as possible. In the end, it returns the total area of the packed items and the position of the last packed item. A complete description of the open space heuristic is given by Wei et al. (2018).

5. Computational experiments

The performance of the proposed VNS method is evaluated through computational experiments using benchmark and newly created instances. We compare our method with state-of-the-art methods, considering the best-known solutions reported in the literature for the G2L-SDVRP and 2L-CVRP. The VNS was coded in C++ and uses the Gurobi Optimizer, version 8.1, to solve the set partitioning model, and the Constraint Programming in the IBM ILOG CPLEX Optimization Studio, version 12.8, to solve the constraint programming model. All experiments were run on a computer with an Intel Core i7-8700 3.2 GHz processor, 8 GB of RAM, and Linux Ubuntu 18.04 LTS as the operating system. We run the proposed VNS 10 times for each instance, with the seed varying from 1 to 10 since it has random internal parameters. From these runs, the value of the best solution found is reported.

5.1. Instances and parameters

We use two sets of instances to evaluate the performance of the proposed VNS. The first set comprises 180 benchmark instances from the 2L-CVRP literature, originally proposed by Iori et al. (2007) and Gendreau et al. (2008). These instances are organized into five classes (Classes 1 to 5) based on the number of rectangular items per customer. Each class has 36 instances in which the number of items per customer is limited to the class number. These instances are available at `http://www.or.deis.unibo.it/`.

The second set (Class 6) includes 36 new instances we generate and use for the first time in this paper. They were generated following the same approach used for generating the instances of Class 5 (see Iori et al. (2007) for more details), except for the number of items per customer, which is in the range $[2, 4]$ instead of $[1, 5]$. Additionally, to define the number of vehicles in the instances, we tried the strategy used by Iori et al. (2007) and Ferreira et al. (2021) but returned infeasible instances. Hence, we decided to set the number of vehicles as the same number in the instances of Class 5. In this way, we were able to guarantee the newly generated instances are feasible. These instances are available at `https://bit.ly/taq`.

Recall that our VNS approach has two input parameters, namely T_{max} and NN. The time limit T_{max} is set according to the number of customers in the instances. If $n \leq 50$, we set T_{max} to 1800 seconds; otherwise, we set it to 3600 seconds, in accordance with Wei et al. (2018). Through preliminary tests, $NN = 100$ provided the best overall results.

5.2. Results of the 2L-CVRP

As mentioned, our VNS method is the first metaheuristic proposed for the G2L-SDVRP. Hence, to assess its performance in relation to other methods in the literature, we first solve the 2L-CVRP instances. Next, we compare our results against the state-of-the-art algorithms for this problem: the VNS$_W$ of Wei et al. (2015) and the SA of Wei et al. (2018). The best-known solution (BKS) is used to verify the quality of the solutions obtained by all the methods. The BKS is obtained from these authors. For each method, we report the cost of the best solution obtained from 10 runs.

Table 1 presents a comparison of the VNS results with the literature on the pure CVRP instances (Class 1) and 2L-CVRP (average over Classes 2–5). For each method, the table shows the number of worse, equal, and better solutions compared to the BKS; the relative difference (Gap) in percentage, computed as $100 \times ((f_{VNS} - f_{BKS})/f_{BKS})$, where f_{VNS} is the value of the best solution obtained using the VNS and f_{BKS} is the BKS value; and the average computing time in seconds. The computing time refers to the time until obtaining the last best solution, which is in accordance with Wei et al. (2015, 2018). We did not compare computing times because the computer configurations (i.e., CPU speed, operating system, compiler, among others) are different, and it could result in an unfair comparison. The detailed results of the 2L-CVRP obtained using the proposed VNS are available in Appendix A, Table A.6.

The results show that our VNS is competitive with the state-of-the-art methods and has the smallest gap value overall. For Class 1, all methods obtained more than 50% of the solutions equal to the BKS. Besides that, the VNS improved the solution of five instances. Since all customers in Class 1 demand only one item of dimensions (1,1), only the routing counterpart is examined in these instances. Therefore, these results indicate that the routing components of our VNS are very efficient. In Classes 2-5, the proposed approach obtained 17 solutions better than the BKS, with an average improvement of 0.09%.

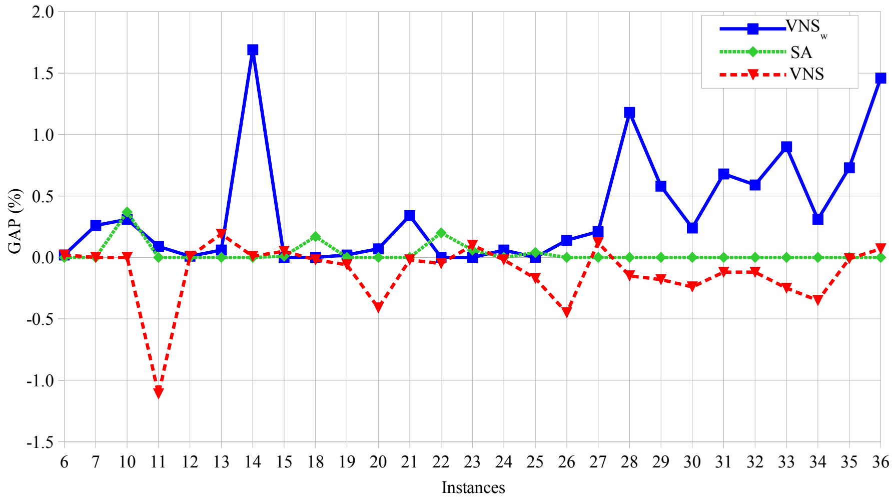
In Figure 5, we present the average gap of Classes 2 to 5 for each instance in which the solution of one method differs from the BKS. The figure shows that VNS$_W$ obtains the most distant solutions

Table 1: Results obtained using the proposed VNS and the state-of-the-art methods on instances of the 2L-CVRP.

Metaheuristic	Class 1					Classes 2–5				
	Worse	Equal	Better	Gap (%)	Time (s)	Worse	Equal	Better	Gap (%)	Time (s)
VNS_w	8	28	0	0.04	460.53	22	14	0	0.28	975.32
SA	7	29	0	0.02	448.63	6	30	0	0.02	1062.95
VNS	4	27	5	0.01	55.49	12	7	17	-0.09	1033.65

from the BKS, and, for three instances, the gap is greater than 1%. The SA approach has a gap varying between 0 and 0.5%. The proposed VNS has no gap greater than 0.5% and obtains a solution better than the BKS with a difference larger than 1%.

Figure 5: Gaps obtained using the proposed VNS and the state-of-the-art methods in instances of Classes 2-5.



For each class, Table 2 has the comparison of the proposed VNS with the BKS. It also shows the average computing time per class. We observe that the larger the number of items, the higher the computing times are. The method achieves the highest average computing time for Class 5, with an average value of 1195.64 seconds. Notably, the VNS has more difficulty solving the instances in Class 5, where the number of items per customer is the highest. This feature makes it more difficult to pack items, as accommodating many items requires efficient utilization of the vehicle base. Overall, the VNS finds better solutions for 50 instances and matches the best solutions for 97 ones. The average improvement over the BKS is 0.04%. It is important to mention that our objective is not to solve the 2L-CVRP, but even so, the proposed VNS is much better compared with the state-of-the-art methods for the 2L-CVRP.

5.3. Results of the G2L-SDVRP

For the G2L-SDVRP, we compare the results obtained using the proposed VNS and those obtained using the branch-and-cut (BC) method in Ferreira et al. (2021). Recall that in the packing procedure, especially in Procedure 6, we pack items over a grid of points, thus reducing the number of points where to pack items. In the preliminary experiments, our method obtained better results when considering, in the constraint programming model, the normal patterns (Herz, 1972) instead

Table 2: Results of the proposed VNS method on each instance class for the 2L-CVRP.

Class	Worse	Equal	Better	Gap (%)	Time (s)
1	4	27	5	0.01	55.49
2	3	24	9	-0.14	676.67
3	0	19	17	-0.18	1072.21
4	3	14	19	-0.25	1190.06
5	23	13	0	0.37	1195.64
1-5	33	97	50	-0.04	838.02

of the meet-in-the-middle patterns (Côté and Iori, 2018), particularly for large-scale instances. In this way, we also extended it to the BC in Ferreira et al. (2021). This means the BC uses the normal patterns when handling the loading subproblems. As a result, an average improvement of 0.03% is obtained using this new version of the BC compared to the original authors. This method is also applied to solve the new instances (Class 6). The complete results obtained using the BC method (with the normal patterns) are presented in Appendix A, Tables A.11 and A.12.

Table 3 summarizes the results obtained using the VNS and BC methods. For the VNS, we present the worst (VNS_{Worst}), average ($VNS_{Average}$) and best (VNS_{Best}) solution values over the ten runs. The first column in the table presents the instance class, and the next two columns show the number of optimal solutions (OPT) and the average computing time (in seconds) for the BC method. Then, for each result of the VNS, we present the number of instances in which the VNS obtained better (B), equal (E), and worse (B) solutions in comparison to the BC; the average relative difference (Gap) between the solution values obtained using the BC and VNS, as a percentage (considering the CO_2 emission); and the worst, average or best computing time (in seconds) for VNS_{Worst}, $VNS_{Average}$ and VNS_{Best}, respectively. Negative values of the gap indicate that the VNS outperforms the BC regarding the solution quality; null values mean that both approaches have the same solution; and values greater than zero indicate the BC method is superior to the VNS. Instances of Class 1 are not included in this experiment since split delivery does not apply to them, given that all customers demand only a single item.

Table 3: Results of the VNS and BC methods in instances of the G2L-SDVRP.

Class	BC		VNS_{Worst}					$VNS_{Average}$					VNS_{Best}				
	OPT	Time (s)	B	E	W	Gap(%)	Time (s)	B	E	W	Gap(%)	Time (s)	B	E	W	Gap(%)	Time (s)
2	5	2279.87	4	7	1	-0.86	105.41	4	7	1	-0.88	74.36	4	8	0	-0.93	1.42
3	4	2572.07	5	5	2	-0.16	133.62	6	4	2	-0.33	143.45	6	6	0	-0.50	15.16
4	6	2264.64	2	6	4	-0.04	297.51	3	5	4	-0.15	266.85	3	6	3	-0.26	119.88
5	5	2383.10	4	5	3	-0.37	292.48	4	5	3	-0.46	247.14	5	6	1	-0.68	162.38
2-5	20	2374.92	15	23	10	-0.36	207.25	17	21	10	-0.46	182.95	18	26	4	-0.59	74.71

The results in Table 3 indicate the superior performance of the VNS approach in relation to the BC method in all classes, regarding both the gap and computing time. The solutions obtained using the VNS are superior considering the VNS_{Worst}, $VNS_{Average}$ and VNS_{Best}, with an average gap of 0.36%, 0.46% and 0.59%, respectively. As expected, the BC method requires more run time than the VNS in all classes, with an average difference larger than 2000 seconds. Moreover, from the detailed results, we observe the BC reports an optimal solution in all instances, and the VNS finds a solution with the same amount of CO_2 emissions. Moreover, the best results obtained using our VNS (VNS_{Best}) show reductions in the CO_2 emissions for 17 instances. It is worse than the BC only in four instances.

Figure 6 compares the quality of the solutions obtained with the VNS against those obtained using the BC, considering the measures VNS_{Worst}, $VNS_{Average}$ and VNS_{Best}. These results show that our VNS outperforms the BC method considering all measures. On average, the VNS can reduce route costs and CO_2 emissions compared to the exact method, highlighting its efficiency in solving the problem. In the worst case, the VNS reduces the CO_2 emissions and route costs by 0.36% and 0.29%, respectively, while in the best case, the gains in reducing the CO_2 emissions and route costs can reach 0.59% and 0.84%.

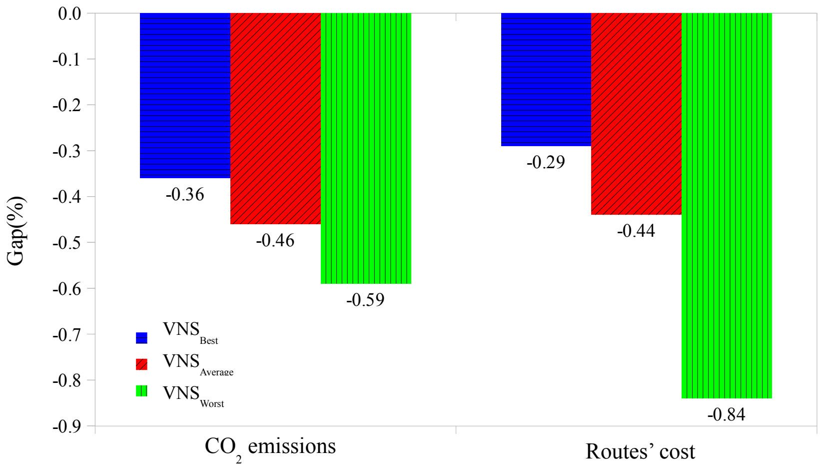


Figure 6: Comparison of the BC solutions with the VNS solutions.

Figure 7 reports the average gap for the worst, average, and best solutions considering the CO_2 emissions (Figure 7a) and route costs (Figure 7b). We calculate the average value for each measure considering the four classes (2–5). Then, we compute the gap in relation to the solution of the BC. Notably, the gain obtained with the VNS regarding CO_2 emissions varies between 0.12% to 4.59%, while the savings regarding route costs are between 0.15% and 6.49%. The worst solution concerns instance E026-08m, which emitted 0.70% more CO_2 than the solution obtained with the BC.

Figure 7: Average gap of the solutions.

(a) CO_2 emissions: GAP_G. (b) Route cost: GAP_R.

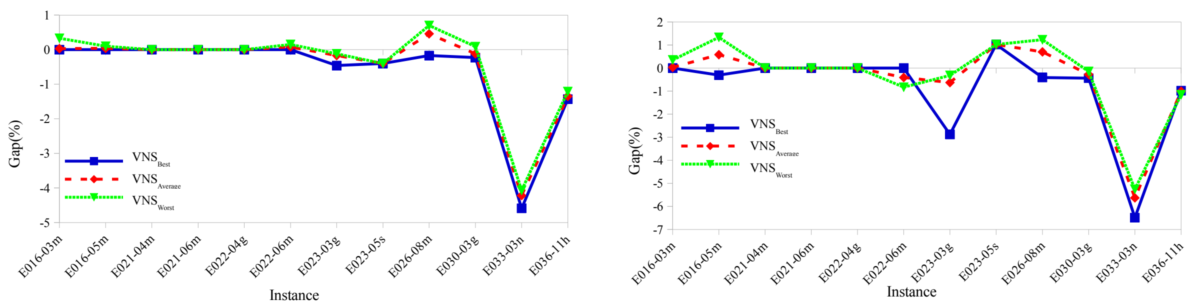


Table 4 presents the solution gaps for instances in which the best solution obtained using the VNS (VNS_{Best}) is different from the solution obtained using the BC method. The maximum reduction in CO_2 emissions and routes cost is 9.38% and 12.19%, respectively, as observed in instance E033-03n of Class 2. When the VNS obtains a solution with higher CO_2 emissions, it is, at most, 0.26% worse than the BC solution. This is a small increase, especially considering the difference in computing times (see Table 3). Finally, we observe an interesting result in instance E016-05m of Class 5, as the best solutions the VNS obtains (VNS_{best}) has the same CO_2 emission of the BC solution, but the VNS improves the route cost by 1.28%.

Table 4: Gap in instances where the BC and VNS have different solutions.

Instancess	Route costs				CO_2 emissions			
	Class 2	Class 3	Class 4	Class 5	Class 2	Class 3	Class 4	Class 5
E016-05m	-	-	-	-1.28	-	-	-	0.00
E022-06m	-	-	0.00	-	-	-	0.005	-
E023-03g	-1.38	-7.92	-1.98	0.15	-0.24	-0.65	-0.68	-0.25
E023-05s	-	-	0.00	2.85	-	-	0.01	-0.82
E026-08m	-	0.32	-0.97	-1.01	-	-1.11	0.18	0.26
E030-03g	-1.27	0.00	-	-0.44	-0.31	-0.33	-	-0.30
E033-03n	-12.19	-4.04	-0.08	-7.43	-9.38	-1.27	-1.35	-5.09
E036-11h	-1.13	-1.27	-1.52	0.00	-1.21	-1.86	-1.29	-1.35

5.4. Results of the G2L-SDVRP for Class 6

Table 5 reports the comparison between the two methods, BC and VNS, for instances of Class 6. The columns present the routes cost (Sol_R), the amount of CO_2 emission (Sol_G), and the total computing time in seconds. Additionally, for the VNS results, the table shows the average gap between the VNS solutions and the BC solutions, in terms of the total cost of routes (GAP_R) and CO_2 emissions (GAP_G). Notably, the average computing time of the VNS solutions is smaller than that of the BC method by about 2000 seconds. On average, the best solutions of the VNS (VNS_{Best}) reduce the CO_2 emissions and the routes cost by 0.20% and 0.16%, respectively. Concerning CO_2 emissions, these solutions are better in four instances, equal in four others, and worse in three instances. Regarding the worst and average results obtained with the VNS (VNS_{Worst} and

Table 5: Results obtained using the VNS and BC methods for instances of the G2L-SDVRP in Class 6.

Instance	BC			VNS_{Worst}					$VNS_{Average}$					VNS_{Best}				
	Sol_R	Sol_G	Time (s)	Sol_R	Sol_G	Time (s)	Gap_R	Gap_G	Sol_R	Sol_G	Time (s)	Gap_R	Gap_G	Sol_R	Sol_G	Time (s)	Gap_R	Gap_G
E016-03m	284	1152.93	4.40	284	1152.93	9.70	0.00	0.00	284.00	1152.93	8.34	0.00	0.00	284	1152.93	0.01	0.00	0.00
E016-05m	308	1494.37	66.16	312	1494.37	7.73	1.30	0.00	310.80	1494.37	6.04	0.91	0.00	308	1494.37	0.02	0.00	0.00
E021-04m	360	1577.67	818.81	365	1585.04	89.16	1.39	0.47	363.00	1581.98	88.39	0.83	0.27	360	1577.67	1.91	0.00	0.00
E021-06m	427	1967.97	3598.25	427	1967.97	32.72	0.00	0.00	427.00	1967.97	25.79	0.00	0.00	427	1967.97	0.22	0.00	0.00
E022-04g	367	1751.55	158.19	367	1751.55	186.22	0.00	0.00	367.00	1751.55	168.72	0.00	0.00	367	1751.55	0.87	0.00	0.00
E022-06m	473	2311.50	3596.45	471	2315.40	32.13	-0.42	0.17	475.00	2312.01	46.27	0.42	0.02	479	2308.55	2.86	1.27	-0.13
E023-03g	653	2414.24	3581.86	690	2484.63	458.40	5.67	2.92	687.70	2482.20	422.54	5.31	2.81	667	2460.34	314.81	2.14	1.91
E023-05s	653	2414.24	3581.81	690	2484.63	439.34	5.67	2.92	684.20	2476.38	438.70	4.78	2.57	655	2426.45	247.33	0.31	0.51
E026-08m	598	2828.42	3597.31	606	2845.57	60.93	1.34	0.61	606.00	2845.57	46.43	1.34	0.61	606	2845.57	0.23	1.34	0.61
E030-03g	662	2560.66	3596.88	650	2534.70	1256.13	-1.81	-1.01	647.50	2529.49	1415.99	-2.19	-1.22	637	2510.52	609.38	-3.78	-1.96
E033-03n	2439	8981.47	3572.22	2433	8852.62	1488.63	-0.25	-1.43	2424.90	8834.57	1410.89	-0.58	-1.64	2424	8832.43	68.56	-0.62	-1.66
E036-11h	707	3250.60	3299.97	695	3205.64	89.32	-1.70	-1.38	690.80	3199.64	105.55	-2.29	-1.57	689	3196.96	2.87	-2.55	-1.65
Average			2456.03			345.87	0.93	0.27			348.64	0.71	0.16			104.09	-0.16	-0.20

VNS$_{\text{Average}}$), we observe a slightly superior performance of the BC method in these instances regarding the solution quality. However, considering the significant difference between the computing times (superior to 2000 seconds), the VNS will likely obtain better quality solutions if it runs longer. Therefore, the VNS is very competitive in Class 6, presenting significantly shorter computing times even in larger instances.

Figure 8 shows the gap of the best solutions obtained using the VNS (VNS$_{Best}$) for the instances in Class 6, considering only the instances in which the VNS and BC methods have different solutions. The maximum reduction in terms of CO_2 emissions is due to instance E030-03g. In this case, the VNS improved the BC solution by 1.96% and reduced the route cost by 3.78%. The VNS worst solution is in instance E023-03g, with a difference in CO_2 emissions by 1.91%. The highest improvement in the route cost is in instance E030-03g, in which the VNS obtains an improvement of 3.78% compared to the BC solution.

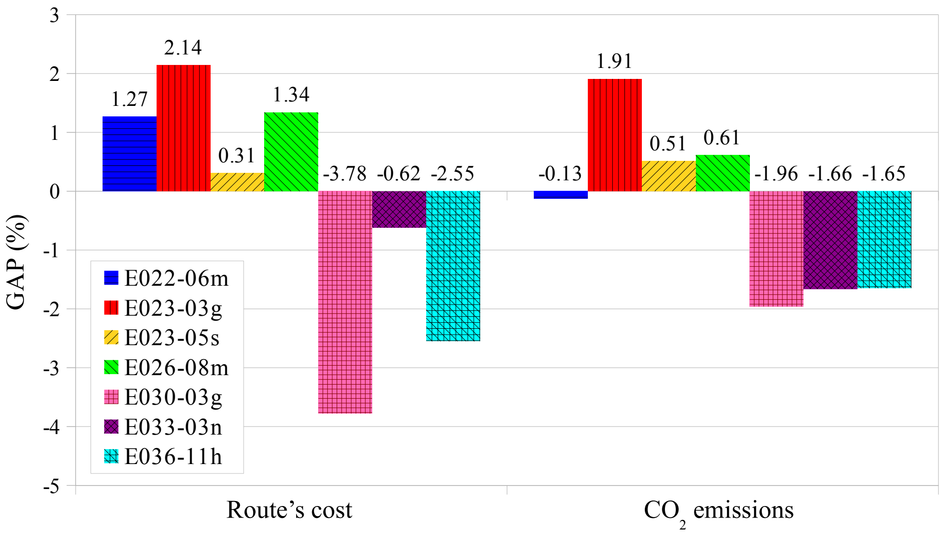


Figure 8: Gap of instances in Class 6 that the VNS and BC have different solutions.

5.5. Convergence analysis and solution improvement on large-scale instances

To further analyze the convergence of the proposed VNS method, we carried out experiments on the large-scale instances 31–36 of Class 5 and Class 6. For each instance, the VNS is executed only once, with the random seed set to 1, to maintain consistency across the experiments. Figures 9 and 10 show the convergence of the solution for these instances in Classes 5 and 6, respectively, representing the reduction of the CO_2 emissions according to the running time. Observing the figures, the VNS shows a rapid convergence in many instances.

As mentioned, the BC method of Ferreira et al. (2021) can only report optimal solutions for small instances. Hence, to verify the efficiency of the proposed VNS on large-scale instances, we compare the final solution obtained with the VNS against its initial solution. The complete results are presented in Appendix A, Tables A.9 and A.10. Figure 11 presents the gap between the initial and final solutions, calculated by $100 \times ((\text{Initial solution} - \text{Best solution})/\text{Best solution})$, considering the best result out of the ten runs (VNS$_{\text{Best}}$). In all classes, we observe an improvement over the initial solutions superior to 30% for more than half the instances. Moreover, the overall improvement is superior to 47%, on average.

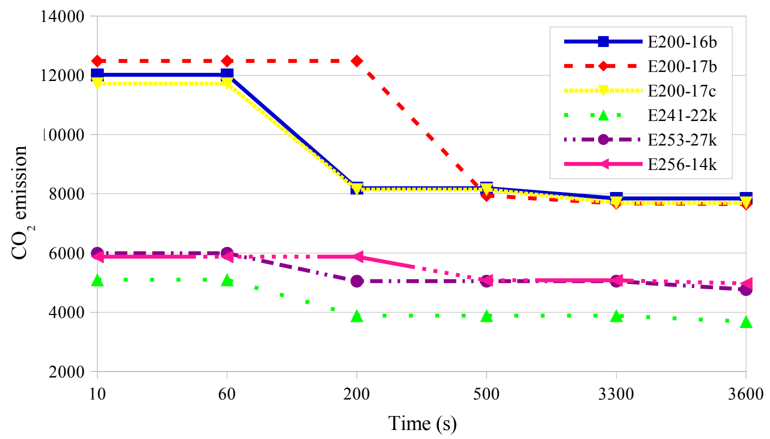


Figure 9: Convergence of the proposed VNS on instances 31–36 of Class 5.

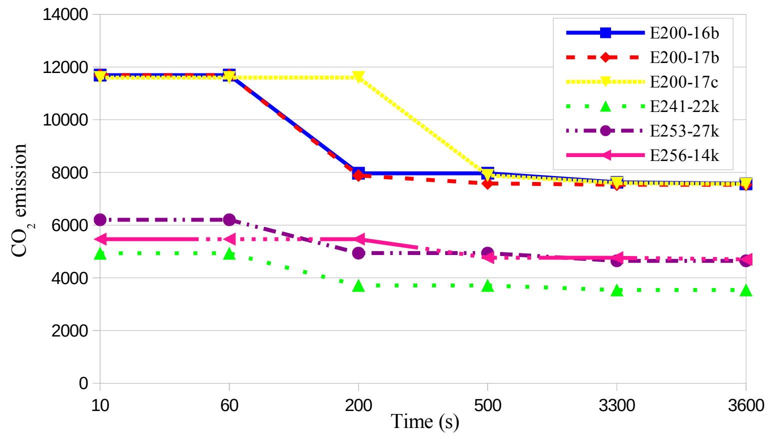


Figure 10: The convergence of the proposed VNS on instances 31–36 of Class 6.

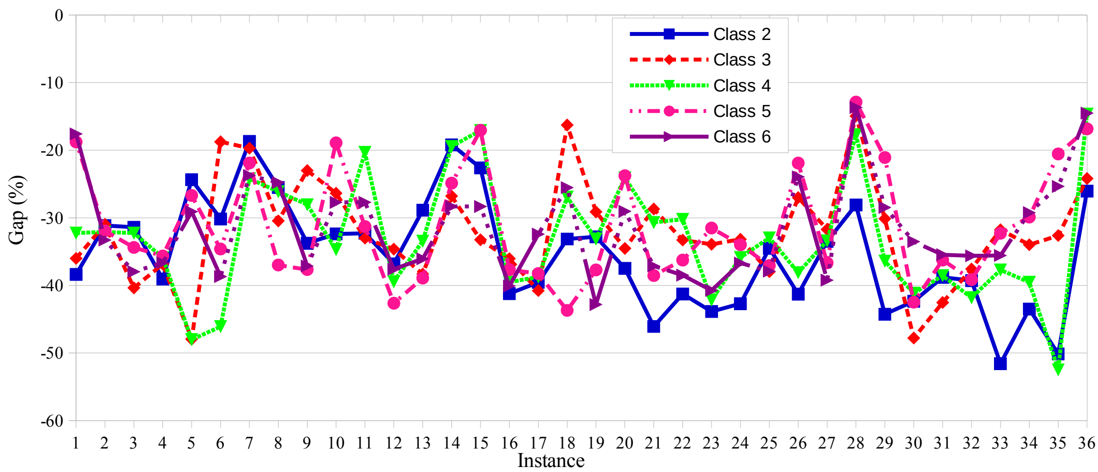
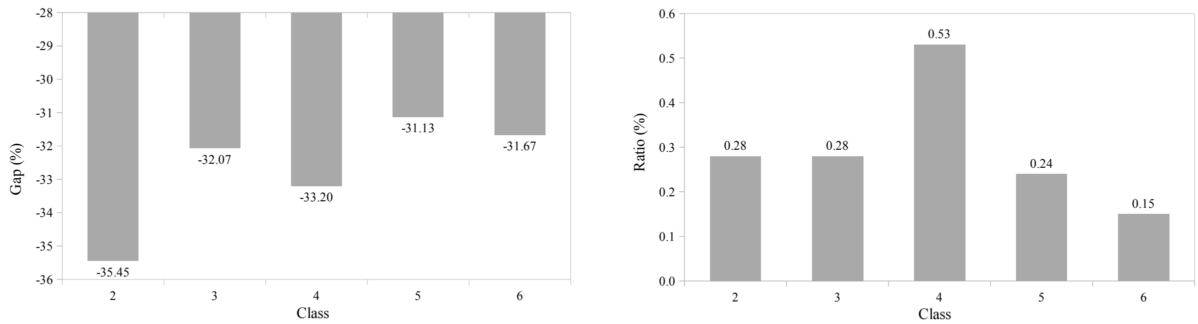


Figure 11: Comparison between the initial and final solutions obtained with the proposed VNS.

Figure 12 illustrates the average gap between the initial and final solutions (Figure 12a) and the ratio between the running time spent at the construction process and the total running time, computed as $100 \times$ (Initial solution time/Final solution time) (Figure 12b). These results show that the proposed VNS has more difficulty improving the initial solutions for instances with more items per customer. Notice the gaps in Classes 5 and 6 have the smallest improvements. In addition, for all classes, the average ratio is smaller than 1%, indicating that the VNS requires low computing time to obtain an initial solution.

Figure 12: Improvement achieved with the VNS compared to its initial solution.

(a) Gap (%). (b) Ratio (%).



6. Concluding remarks

We propose the first metaheuristic method for the green vehicle routing problem with two-dimensional loading constraints and split delivery (i.e., the G2L-SDVRP). Besides defining vehicle routes to supply customers' demand for rectangular items, we need to guarantee the two-dimensional loading of items on each route/vehicle is feasible. Moreover, a customer can be served by one or more vehicles, while the objective aims to minimize CO_2 emissions. The proposed metaheuristic is a variable neighborhood search comprising five neighborhood structures, a local search based on the random variable neighborhood descent, a set partitioning model, a procedure to diversify the search, and different procedures to effectively check the packing feasibility of a route.

The results of the computational experiments for the G2L-SDVRP indicate that the proposed VNS can achieve high-quality solutions compared to other literature methods, particularly the branch-and-cut of Ferreira et al. (2021). On average, the solutions obtained with the VNS reduce the CO_2 emission by 0.38% compared to those obtained with the branch-and-cut method. Furthermore, the computing time required by the VNS to obtain the new, improved solutions is significantly less. Given the 60 instances, the proposed VNS reduces the CO_2 emission for 21 ones and obtains solutions with the same emission for the other 32 instances. For the new instances, we once again confirm the superior performance of the VNS. On average, it obtains improvements superior to 40% compared to the initial solutions.

We also attest to the superior performance of the proposed VNS when solving the capacitated vehicle routing problem with two-dimensional loading constraints (i.e., the 2L-CVRP). Our method is very competitive with the state-of-the-art methods, achieving superior results. It improves the best-known solution in 50 out of 180 instances while obtaining the same solution for the other 97 instances.

There are interesting directions for future research. One trend is to further approximate the problem to the reality of logistics companies by including other practical requirements, e.g., urgent time windows, pickup and delivery, a heterogeneous fleet of vehicles, load-bearing, rotation of items, and cargo stability (Junqueira and Queiroz, 2022). Additionally, one may consider extending the problem to having three-dimensional loading constraints. Another relevant direction is to extend the proposed VNS to handle multi-objective formulations (Queiroz and Mundim, 2020), e.g., in which the route costs and CO_2 emissions are modeled as objectives. Finally, new approaches can be proposed, especially exact techniques that efficiently handle subproblems related to packing and routing decisions, such as branch-and-price and branch-cut-and-price methods (Balster et al., 2023).

Acknowledgments

The authors would like to thank the support from the Coordination for the Improvement of Higher Education Personnel (CAPES) [grant PROEX-10855292/D]; State of São Paulo Research Foundation (FAPESP) [grant numbers 2022/10303-0, 2022/05803-3, 2019/23596-2, 2013/07375-0]; National Council for Scientific and Technological Development (CNPq) [grant numbers 405369/2021-2, 313220/2020-4, 311185/2020-7, 405369/2021-2 and 309161/2022-3]; and State of Goiás Research Foundation (FAPEG).

References

Almeida Cunha, J.G., de Lima, V.L., Queiroz, T.A., 2020. Vehicle routing problems with split deliveries. 4OR - A Quarterly Journal of Operations Research 18, 293–339.
Alvarez-Valdés, R., Parreño, F., Tamarit, J.M., 2009. A branch and bound algorithm for the strip packing problem. OR Spectrum 31, 431–459.
Annouch, A., Bellabdaoui, A., Minkhar, J., 2016. Split delivery and pickup vehicle routing problem with two-dimensional loading constraints, in: 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA), IEEE. pp. 1–6.
Archetti, C., Bianchessi, N., Speranza, M.G., 2014. Branch-and-cut algorithms for the split delivery vehicle routing problem. European Journal of Operational Research 238, 685–698.
Archetti, C., Savelsbergh, M.W., Speranza, M.G., 2006. Worst-case analysis for split delivery vehicle routing problems. Transportation science 40, 226–234.
Archetti, C., Speranza, M.G., 2012. Vehicle routing problems with split deliveries. International transactions in operational research 19, 3–22.
Balster, I., Bulhões, T., Munari, P., Pessoa, A.A., Sadykov, R., 2023. A new family of route formulations for split delivery vehicle routing problems. Transportation Science 57, 1115–1401.
Bektaş, T., Laporte, G., 2011. The pollution-routing problem. Transportation Research Part B: Methodological 45, 1232–1250.
Bortfeldt, A., Yi, J., 2020. The split delivery vehicle routing problem with three-dimensional loading constraints. European Journal of Operational Research 282, 545–558.
Boschetti, M.A., Mingozzi, A., Hadjiconstantinou, E., 2002. New upper bounds for the two-dimensional orthogonal non-guillotine cutting stock problem. IMA Journal of Management Mathematics 13, 95–119.
Ceschia, S., Schaerf, A., Stützle, T., 2013. Local search techniques for a routing-packing problem. Computers & Industrial Engineering 66, 1138–1149.
Clarke, G., Wright, J.W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations research 12, 568–581.
Clautiaux, F., Jouglet, A., Carlier, J., Moukrim, A., 2008. A new constraint programming approach for the orthogonal packing problem. Computers & Operations Research 35, 944–959.
Côté, J.F., Gendreau, M., Potvin, J.Y., 2014. An exact algorithm for the two-dimensional orthogonal packing problem with unloading constraints. Operations Research 62, 1126–1141.

Côté, J.F., Guastaroba, G., Speranza, M.G., 2017. The value of integrating loading and routing. European Journal of Operational Research 257, 89–105.

Côté, J.F., Iori, M., 2018. The meet-in-the-middle principle for cutting and packing problems. INFORMS Journal on Computing 30, 646–661.

Dabia, S., Demir, E., Woensel, T.V., 2016. An exact approach for a variant of the pollution-routing problem. Transportation Science 51, 607–628.

Demir, E., Bektaş, T., Laporte, G., 2014. A review of recent research on green road freight transportation. European journal of operational research 237, 775–793.

Dewi, S.K., Utama, D.M., 2021. A new hybrid whale optimization algorithm for green vehicle routing problem. Systems Science & Control Engineering 9, 61–72.

Dror, M., Laporte, G., Trudeau, P., 1994. Vehicle routing with split deliveries. Discrete Applied Mathematics 50, 239–254.

Ehmke, J.F., Campbell, A.M., Thomas, B.W., 2016. Vehicle routing to minimize time-dependent emissions in urban areas. European Journal of Operational Research 251, 478–494.

Ferreira, H.S., Bogue, E.T., Noronha, T.F., Belhaiza, S., Prins, C., 2018. Variable neighborhood search for vehicle routing problem with multiple time windows. Electronic Notes in Discrete Mathematics 66, 207–214.

Ferreira, K.M., Queiroz, T.A., Toledo, F.M.B., 2021. An exact approach for the green vehicle routing problem with two-dimensional loading constraints and split delivery. Computers & Operations Research 136, 105452.

Gendreau, M., Iori, M., Laporte, G., Martello, S., 2008. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. Networks: An International Journal 51, 4–18.

Golden, B.L., Raghavan, S., Wasil, E.A., 2008. The vehicle routing problem: latest advances and new challenges. volume 43. Springer.

Hemmelmayr, V.C., Doerner, K.F., Hartl, R.F., 2009. A variable neighborhood search heuristic for periodic routing problems. European Journal of Operational Research 195, 791–802.

Herz, J.C., 1972. Recursive computational procedure for two-dimensional stock cutting. IBM Journal of Research and Development 16, 462–469.

Imran, A., Salhi, S., Wassan, N.A., 2009. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research 197, 509–518.

Iori, M., Salazar-González, J.J., Vigo, D., 2007. An exact approach for the vehicle routing problem with two-dimensional loading constraints. Transportation Science 41, 253–264.

Ji, B., Zhou, S., Samson, S.Y., Wu, G., 2021. An enhanced neighborhood search algorithm for solving the split delivery vehicle routing problem with two-dimensional loading constraints. Computers & Industrial Engineering 162, 107720.

Junqueira, L., Queiroz, T.A., 2022. The static stability of support factor-based rectangular packings: an assessment by regression analysis. International Transactions in Operational Research 29, 574–599.

Krebs, C., Ehmke, J.F., 2023. Solution validator and visualizer for (combined) vehicle routing and container loading problems. Annals of Operations Research 326, 561–579.

Li, X., Yuan, M., Chen, D., Yao, J., Zeng, J., 2018. 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, pp. 528–536.

Lin, C., Choy, K.L., Ho, G.T., Chung, S.H., Lam, H., 2014. Survey of green vehicle routing problem: past and future trends. Expert Systems with Applications 41, 1118–1138.

Martello, S., Toth, P., 1990. Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc.

Matos, M.R.S., Frota, Y., Ochi, L.S., 2018. Green vehicle routing and scheduling problem with split delivery. Electronic Notes in Discrete Mathematics 69, 13–20.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers & Operations Research 24, 1097–1100.

Munari, P., Savelsbergh, M., 2020. A column generation-based heuristic for the split delivery vehicle routing problem with time windows, in: SN Operations Research Forum, Springer. pp. 1–24.

Munari, P., Savelsbergh, M., 2022. Compact formulations for split delivery routing problems. Transportation Science 56, 1022–1043.

Nascimento, O.X., Queiroz, T.A., Junqueira, L., 2021. Practical constraints in the container loading problem: Comprehensive formulations and exact algorithm. Computers & Operations Research 128, 105186.

Penna, P.H.V., Subramanian, A., Ochi, L.S., 2013. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. Journal of Heuristics 19, 201–232.

Pollaris, H., Braekers, K., Caris, A., Janssens, G.K., Limbourg, S., 2015. Vehicle routing problems with loading

constraints: state-of-the-art and future directions. OR Spectrum 37, 297–330.
Queiroz, T.A., Mundim, L.R., 2020. Multiobjective pseudo-variable neighborhood descent for a bicriteria parallel machine scheduling problem with setup time. International Transactions in Operational Research 27, 1478–1500.
Rajaei, M., Moslehi, G., Reisi-Nafchi, M., 2022. The split heterogeneous vehicle routing problem with three-dimensional loading constraints on a large scale. European Journal of Operational Research 299, 706–721.
Sadati, M.E.H., Çatay, B., 2021. A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review 149, 102293.
Salimifard, K., Shahbandarzadeh, H., Raeesi, R., 2012. Green transportation and the role of operation research, in: Int. Conf. Traffic Transp. Eng.(ICTTE 2012), IACSIT Press. pp. 74–79.
Shi, J., Zhang, J., Wang, K., Fang, X., 2018. Particle swarm optimization for split delivery vehicle routing problem. Asia-Pacific Journal of Operational Research 35, 1840006.
Silva, L.C., Queiroz, T.A., Toledo, F.M.B., 2022. Integer formulations for the integrated vehicle routing problem with two-dimensional packing constraints. Pesquisa Operacional 42, e248686.
Silva, M.M., Subramanian, A., Ochi, L.S., 2015. An iterated local search heuristic for the split delivery vehicle routing problem. Computers & Operations Research 53, 234–249.
Subramanian, A., Battarra, M., 2013. An iterated local search algorithm for the travelling salesman problem with pickups and deliveries. Journal of the Operational Research Society 64, 402–409.
Subramanian, A., Drummond, L.M., Bentes, C., Ochi, L.S., Farias, R., 2010. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research 37, 1899–1911.
Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods and Applications. Second ed., MOS-SIAM Series in Optimization.
Vornhusen, B., Kopfer, H., 2015. Emission vehicle routing problem with split delivery and a heterogeneous vehicle fleet, in: International Conference on Computational Logistics, Springer. pp. 76–90.
Wei, L., Zhang, Z., Zhang, D., Leung, S.C., 2018. A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. European Journal of Operational Research 265, 843–859.
Wei, L., Zhang, Z., Zhang, D., Lim, A., 2015. A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. European Journal of Operational Research 243, 798–814.
Xiao, Y., Konak, A., 2016. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. Transportation Research Part E: Logistics and Transportation Review 88, 146–166.
Xiao, Y., Zhao, Q., Kaku, I., Xu, Y., 2012. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Computers & Operations Research 39, 1419–1431.
Yi, J., Bortfeldt, A., 2018. The capacitated vehicle routing problem with three-dimensional loading constraints and split delivery—a case study, in: Operations Research Proceedings 2016: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Helmut Schmidt University Hamburg, Germany, August 30-September 2, 2016, Springer. pp. 351–356.
Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2013. Integrated distribution and loading planning via a compact metaheuristic algorithm. European Journal of Operational Research 228, 56–71.
Zhang, S., Lee, C., Choy, K., Ho, W., Ip, W., 2014. Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. Transportation Research Part D: Transport and Environment 31, 85–99.
Zhang, X., Chen, L., Gendreau, M., Langevin, A., 2022. A branch-and-cut algorithm for the vehicle routing problem with two-dimensional loading constraints. European Journal of Operational Research 302, 259–269.

Appendix A. Detailed results of the computational experiments

Table A.6 presents the detailed results obtained using the proposed VNS method for the 2L-CVRP. For each class, the following information is given: instance name; the value of the best-known solution (BKS) in the literature; the value of the solution obtained with the VNS, the computing time to obtain the best solution, in seconds (T_B); the total computing time of the VNS (T_T), in seconds; and, the relative difference (Gap) between the solution value (f_{Sol}) and the BKS f_{BKS}, computed as $100 \times ((f_{Sol} - f_{BKS})/f_{BKS})$.

Table A.7 and A.8 show the detailed results obtained using the proposed VNS for each instance of Classes 2 to 6. This table presents the instance name; the number of customers (n); and, for the VNS, we present the worst ($\text{VNS}_{\text{Worst}}$), average ($\text{VNS}_{\text{Average}}$) and best ($\text{VNS}_{\text{Best}}$) solutions over the ten runs. We also present the route costs ($\mathrm{Sol_R}$) and the amount of CO_2 emitted ($\mathrm{Sol_G}$); and the computing time, in seconds, to obtain the best solution.

Table A.9 and A.10 have a comparison between the VNS final solutions and the VNS initial solutions. The following information is presented for each instance: initial solution value (Sol_I); final solution value (Sol_F); the gap between the final and initial solutions, computed as $100 \times (Sol_I - Sol_F)/Sol_F$; computing time, in seconds, to obtain the initial (T_I) and final (T_F) solutions; and, the difference between the final and initial computing times (RT - computed as $100 \times (T_I/T_F)$).

Tables A.11 and A.12 show the detailed results obtained using the branch-and-cut (BC) method of Ferreira et al. (2021) (with the normal patterns) for the G2L-SDVRP. For each instance, these tables present the instance name, the number of customers, and the number of items, the lower (K_{min}) and upper (K_{max}) bounds on the number of vehicles to serve all customers' demands, the number of vehicles in the solution (VH), the number of customers with split delivery in the solution (CS), the routes cost (Sol_R), the amount of CO_2 emission (Sol_G), the total computing time ($Time_T$), the computing time for solving the packing subproblems ($Time_P$), and the number of cuts related to infeasible packings (Cut_P).

Table A.6: Results obtained with the VNS for the 2L-CVRP instances using exactly K_{max} vehicles and not allowing routes with a single customer.

Inst.	Class 1					Class 2					Class 3					Class 4					Class 5				
	BKS	Solution	T_B	T_T	Gap(%)	BKS	Solution	T_B	T_T	Gap(%)	BKS	Solution	T_B	T_T	Gap(%)	BKS	Solution	T_B	T_T	Gap(%)	BKS	Solution	T_B	T_T	Gap(%)
1	278.73	278.73	0.00	0.88	0.00	290.84	290.84	0.50	26.47	0.00	284.52	284.52	0.94	39.81	0.00	294.25	294.25	0.25	15.23	0.00	278.73	278.73	11.67	44.98	0.00
2	334.96	334.96	0.00	0.49	0.00	347.73	347.73	0.02	1.79	0.00	352.16	352.16	0.05	5.12	0.00	342.00	342.00	0.00	3.38	0.00	334.96	334.96	0.00	0.74	0.00
3	358.40	358.40	0.00	1.05	0.00	403.93	403.93	0.03	16.11	0.00	394.72	394.72	1.80	31.51	0.00	368.56	368.56	1.44	32.14	0.00	358.40	358.40	0.11	12.77	0.00
4	430.88	430.89	0.00	0.94	0.00	440.94	440.94	0.05	4.92	0.00	440.68	440.68	0.35	8.12	0.00	447.37	447.37	1.15	17.19	0.00	430.88	430.89	0.13	11.76	0.00
5	375.28	375.28	0.00	1.36	0.00	388.72	388.72	0.78	29.92	0.00	381.69	381.69	0.62	22.84	0.00	383.87	383.88	2.99	47.24	0.00	375.28	375.28	0.01	14.77	0.00
6	495.85	495.85	0.00	1.77	0.00	499.08	499.08	0.10	8.10	0.00	504.68	504.68	1.54	15.23	0.00	498.32	498.32	0.25	35.89	0.00	495.85	495.85	0.26	6.95	0.00
7	568.56	568.56	0.00	5.95	0.00	734.65	734.65	0.11	23.47	0.00	702.59	702.59	13.27	80.18	0.00	703.49	703.49	15.08	229.83	0.00	658.64	658.64	4.89	192.17	0.00
8	568.56	568.56	0.00	5.02	0.00	725.91	725.91	2.55	29.43	0.00	741.12	741.12	0.38	48.07	0.00	697.92	697.92	11.01	132.72	0.00	621.85	621.85	11.25	429.97	0.00
9	607.65	607.65	0.00	1.83	0.00	611.49	611.49	0.01	7.82	0.00	613.90	613.90	0.10	18.39	0.00	625.10	625.10	1.37	18.37	0.00	607.65	607.65	0.01	9.53	0.00
10	535.74	535.80	0.00	7.68	0.01	700.20	700.20	1.72	89.78	0.00	628.94	628.94	10.02	223.30	0.00	715.82	715.82	125.74	405.82	0.00	690.96	691.04	81.89	646.95	0.01
11	505.01	505.01	0.00	5.60	0.00	721.54	721.54	1.28	110.38	0.00	717.37	717.37	7.25	124.08	0.00	815.68	793.07	719.07	1036.21	-2.77	624.82	624.82	56.08	861.57	0.00
12	610.00	610.00	0.19	4.45	0.00	619.63	619.63	1.48	36.27	0.00	610.00	610.00	0.23	7.68	0.00	618.23	618.23	17.00	135.14	0.00	610.00	610.23	0.52	44.44	0.04
13	2006.34	2006.34	0.00	11.20	0.00	2669.39	2669.39	1.47	108.09	0.00	2486.44	2486.44	25.64	228.91	0.00	2609.36	2609.36	306.47	703.08	0.00	2416.04	2434.99	19.99	971.81	0.78
14	837.67	837.67	0.00	85.76	0.00	1092.51	1092.51	136.15	461.17	0.00	1039.06	1037.59	607.97	1008.23	-0.14	982.25	981.95	4.10	454.71	-0.03	922.75	925.04	99.29	1800.15	0.25
15	837.67	837.67	0.00	97.59	0.00	1041.75	1044.78	792.79	1102.33	0.29	1181.68	1181.68	29.10	633.34	0.00	1246.49	1245.90	460.81	1381.20	-0.05	1229.95	1230.37	26.81	1656.77	0.03
16	698.61	698.61	0.08	6.10	0.00	698.61	698.61	0.03	14.88	0.00	698.61	698.61	0.29	18.21	0.00	708.20	708.20	7.39	58.25	0.00	698.61	698.61	0.05	11.33	0.00
17	861.79	861.79	0.09	8.50	0.00	870.86	870.86	0.23	13.02	0.00	861.79	861.79	0.15	10.16	0.00	861.79	861.79	0.11	17.07	0.00	861.79	861.79	0.21	11.04	0.00
18	723.54	723.54	0.10	16.52	0.00	1053.09	1059.44	35.46	453.78	0.60	1102.17	1102.17	792.22	1316.89	0.00	1134.11	1128.23	623.12	1673.68	-0.52	926.34	926.34	37.95	1800.68	0.00
19	524.61	524.61	0.12	11.40	0.00	792.07	792.07	44.46	340.55	0.00	801.13	801.13	40.87	460.27	0.00	801.21	799.31	74.44	848.09	-0.24	652.15	652.15	131.73	1800.22	0.00
20	241.97	241.97	0.11	22.88	0.00	545.68	545.68	813.14	1616.73	0.00	541.58	536.71	1558.69	1801.20	-0.90	551.72	549.78	1637.66	1801.87	-0.35	478.15	478.77	1764.84	1801.24	0.13
21	687.60	687.60	0.18	69.15	0.00	1060.72	1060.72	426.97	1155.26	0.00	1149.90	1149.20	266.39	1364.35	-0.06	1000.25	990.44	1095.04	1800.60	-0.98	886.00	895.51	1630.67	1802.45	1.07
22	740.66	740.66	0.39	41.10	0.00	1081.44	1081.44	103.13	646.55	0.00	1094.66	1094.16	888.91	1801.30	-0.05	1089.27	1079.27	524.48	1801.83	-0.92	948.60	956.86	1462.27	1804.33	0.87
23	835.26	835.26	5.92	57.59	0.00	1093.27	1093.27	150.43	911.29	0.00	1117.54	1117.09	1076.72	1801.99	-0.04	1093.01	1092.82	267.14	1650.96	-0.02	948.68	955.05	1680.08	1801.34	0.67
24	1024.69	1024.69	4.85	52.36	0.00	1222.43	1222.43	92.77	470.87	0.00	1118.44	1118.44	92.19	702.21	0.00	1141.97	1137.38	1138.31	1801.36	-0.40	1046.08	1049.58	317.39	1382.69	0.33
25	826.14	826.14	0.99	135.96	0.00	1453.98	1453.98	331.01	1163.64	0.00	1433.92	1425.18	2709.13	3600.00	-0.61	1435.18	1432.50	3269.58	3600.00	-0.19	1183.63	1192.91	3454.88	3600.00	0.78
26	819.56	819.56	0.03	44.42	0.00	1323.23	1309.65	1660.81	2544.72	-1.03	1392.43	1389.60	3357.81	3600.00	-0.20	1447.03	1437.73	2285.40	3600.00	-0.64	1252.65	1254.60	2635.37	3600.00	0.16
27	1082.65	1082.65	1.79	165.72	0.00	1367.85	1367.85	412.15	1586.10	0.00	1423.74	1422.34	583.42	2403.14	-0.10	1353.06	1347.31	1868.41	3600.00	-0.42	1259.17	1277.67	2708.90	3600.00	1.47
28	1040.70	1042.12	0.27	52.16	0.14	2632.55	2635.74	2050.61	3600.00	0.12	2737.42	2723.20	2701.26	3600.00	-0.52	2690.69	2682.01	2574.80	3600.00	-0.32	2399.25	2405.90	2178.04	3600.00	0.28
29	1162.96	1162.96	0.56	106.55	0.00	2285.84	2278.61	2125.55	3600.00	-0.32	2150.35	2145.46	2828.70	3600.00	-0.23	2299.32	2294.56	3183.75	3600.00	-0.21	2179.12	2179.64	3103.80	3600.00	0.02
30	1028.42	1028.42	44.78	182.32	0.00	1875.38	1865.82	2025.00	3600.00	-0.51	1912.09	1905.34	1820.33	3600.00	-0.35	1904.42	1898.98	2878.16	3600.00	-0.29	1565.96	1572.38	3596.81	3600.00	0.41
31	1299.56	1318.29	55.53	156.90	1.44	2341.08	2329.05	2283.00	3600.00	-0.51	2354.21	2341.01	3398.89	3600.00	-0.56	2459.59	2451.07	2836.66	3600.00	-0.35	2053.57	2076.61	3444.84	3600.00	1.12
32	1296.91	1291.50	433.76	3600.00	-0.42	2365.99	2343.18	1774.55	3600.00	-0.96	2320.35	2304.94	3358.95	3600.00	-0.66	2343.29	2339.51	2983.86	3600.00	-0.16	2016.58	2048.01	2067.06	3600.00	1.56
33	1296.13	1291.50	300.74	3119.32	-0.36	2349.98	2337.33	1204.12	3600.00	-0.54	2447.20	2433.33	3501.94	3600.00	-0.57	2446.05	2446.05	3510.56	3600.00	0.00	2044.88	2048.55	2808.43	3600.00	0.18
34	708.39	707.81	265.76	3600.00	-0.08	1217.24	1209.02	3211.11	3600.00	-0.68	1249.07	1239.84	2244.98	3600.00	-0.74	1241.13	1233.99	3575.56	3600.00	-0.58	1062.18	1070.71	2573.87	3600.00	0.80
35	862.79	859.29	198.63	3600.00	-0.41	1434.99	1422.06	3080.15	3600.00	-0.90	1511.66	1504.65	3272.87	3600.00	-0.46	1550.24	1557.06	3585.61	3600.00	0.44	1278.90	1291.33	3533.00	3600.00	0.97
36	583.98	583.38	682.77	3600.00	-0.10	1755.33	1743.55	1596.51	3600.00	-0.67	1833.97	1827.40	3405.61	3600.00	-0.36	1713.71	1715.55	3255.55	3600.00	0.11	1541.07	1562.70	3600.00	3600.00	1.40
Average			55.49	524.46	0.01			676.67	1260.37	-0.14			1072.21	1493.74	-0.18			1190.06	1647.27	-0.25			1195.64	1725.57	0.37

Table A.7: Results of the VNS method for the G2L-SDVRP instances in Classes 2 to 5.

Instance		Class 2									Class 3									Class 4									Class 5								
		VNS_{Best}			$VNS_{Average}$			VNS_{Worst}			$VNS_{Average}$			VNS_{Worst}			$VNS_{Average}$			VNS_{Worst}			$VNS_{Average}$			VNS_{Worst}											
Name	n	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)
E016-03m	15	282	1329.64	47.43	282.00	1329.64	58.54	282	1329.64	71.07	291	1345.90	98.66	291.40	1347.58	142.44	295	1362.69	120.44	288	1195.44	20.72	288.00	1195.44	27.66	288	1195.44	38.25	284	1152.93	26.85	284.00	1152.93	34.83	284	1152.93	39.85
E016-05m	15	330	1558.24	5.45	332.40	1558.24	6.72	334	1558.24	8.08	333	1558.88	13.71	336.50	1561.20	20.65	342	1564.90	13.11	308	1494.37	8.97	310.40	1494.37	11.02	312	1494.37	13.92	308	1494.37	4.57	311.20	1494.37	6.89	312	1494.37	9.83
E021-04m	20	402	1683.62	23.27	402.00	1683.62	30.82	402	1683.62	40.52	391	1638.97	44.59	391.00	1638.97	60.87	391	1638.97	78.98	380	1605.27	45.35	380.00	1605.27	59.70	380	1605.27	80.49	360	1577.67	20.63	360.00	1577.67	29.05	360	1577.67	39.20
E021-06m	20	443	2022.97	12.64	443.00	2022.97	15.84	443	2022.97	17.98	427	1966.85	19.86	427.00	1966.85	22.06	427	1966.85	26.39	436	1995.89	34.15	436.00	1995.89	41.42	436	1995.89	51.71	436	2002.84	27.99	436.00	2002.84	31.34	436	2002.84	40.76
E022-04g	21	382	1811.93	47.94	382.00	1811.93	65.30	382	1811.93	87.31	373	1769.66	44.60	373.00	1769.66	62.87	373	1769.66	76.39	377	1825.55	87.36	377.00	1825.55	124.82	377	1825.55	155.83	367	1751.55	20.79	367.00	1751.55	31.91	367	1751.55	44.98
E022-06m	21	473	2326.14	18.81	473.00	2326.14	24.88	473	2326.14	30.87	499	2338.15	27.63	499.00	2338.15	32.33	499	2338.15	43.29	479	2308.89	167.82	471.80	2315.88	119.95	471	2316.66	146.41	479	2308.77	23.25	478.20	2309.38	31.02	471	2314.88	27.97
E023-03g	22	715	2547.63	49.22	715.00	2547.63	58.83	715	2547.63	79.01	674	2502.23	154.11	726.20	2517.05	107.88	732	2518.70	130.10	694	2498.22	287.30	705.10	2511.91	179.29	708	2515.22	202.56	651	2437.96	162.63	651.00	2437.96	228.52	651	2437.96	295.85
E023-05s	22	681	2464.52	39.66	681.00	2464.52	64.22	681	2464.52	96.75	750	2564.96	72.81	750.00	2564.96	108.29	750	2564.96	141.10	699	2494.76	121.42	699.00	2494.76	224.33	699	2494.76	457.16	650	2393.99	166.86	650.00	2393.99	253.53	650	2393.99	317.36
E026-08m	25	613	2879.20	37.28	610.60	2893.90	28.61	610	2897.58	28.67	621	2885.36	135.23	623.40	2903.54	91.89	632	2914.50	53.64	615	2886.15	116.86	622.40	2891.41	89.73	623	2897.80	59.26	588	2797.82	65.10	607.80	2832.03	71.31	612	2838.17	78.84
E030-03g	29	700	2663.79	160.06	700.00	2663.79	234.52	700	2663.79	430.06	622	2503.73	282.77	621.80	2503.74	502.50	620	2503.87	327.06	703	2657.68	417.97	705.00	2660.16	560.02	707	2669.31	522.52	686	2625.00	673.49	688.50	2634.87	669.02	692	2647.20	608.92
E030-04s	29	709	2734.84	156.12	711.90	2735.70	225.30	738	2743.47	151.65	705	2674.84	462.84	705.00	2674.98	298.92	705	2675.00	369.85	782	2894.96	1131.61	794.50	2927.23	941.49	806	2947.58	714.15	630	2483.86	1126.28	639.60	2523.35	1216.21	639	2532.66	1352.07
E031-09h	30	609	2876.08	78.97	611.30	2876.83	112.38	632	2883.60	84.35	588	2794.28	34.46	594.70	2798.32	47.79	605	2803.45	33.71	606	2844.75	332.41	606.40	2849.96	341.51	608	2866.04	242.82	592	2762.39	146.49	591.00	2763.91	221.98	588	2773.15	169.62
E033-03n	32	2693	9528.91	138.14	2693.60	9535.02	243.28	2702	9543.50	288.57	2472	9101.12	332.73	2506.30	9140.82	483.96	2539	9167.02	482.97	2621	9269.25	1800.00	2618.20	9270.18	1483.82	2592	9273.24	1693.45	2381	8787.90	1800.00	2442.40	8882.08	1477.11	2468	8901.17	1800.00
E033-04g	32	1130	4718.85	288.97	1129.40	4720.52	440.11	1124	4735.52	525.67	1026	4519.54	1661.64	1068.50	4587.15	1334.75	1080	4606.52	1051.38	979	4361.11	1447.64	982.40	4367.54	1564.36	993	4373.05	1180.99	934	4204.27	1800.00	940.40	4250.22	1800.00	960	4299.03	1800.00
E033-05s	32	1041	4548.26	612.03	1057.30	4594.23	626.23	1098	4624.68	311.39	1168	4823.86	446.09	1168.00	4823.86	751.95	1168	4823.86	1096.45	1235	4921.60	1800.00	1231.30	4926.32	1800.00	1227	4935.93	1800.00	1227	4912.87	1800.00	1225.30	4922.27	1800.00	1237	4930.69	1800.00
E036-11h	35	699	3229.32	52.49	699.00	3229.32	60.76	699	3229.32	86.07	702	3230.41	119.36	703.20	3232.85	85.66	704	3233.30	109.92	711	3230.45	236.74	708.90	3238.27	280.42	704	3255.78	148.55	708	3230.17	66.92	708.00	3230.17	101.17	708	3230.17	206.14
E041-14h	40	848	3981.83	72.16	850.80	3986.67	78.91	855	3993.93	76.52	847	3972.59	84.22	832.60	3975.10	103.65	829	3975.73	118.86	846	3976.00	103.65	845.50	3977.75	189.39	841	3993.48	81.13	836	3972.50	114.77	831.90	3973.00	198.55	829	3973.49	303.30
E045-04f	44	1048	3620.90	1722.59	1054.00	3634.19	1696.05	1064	3658.51	1617.70	1085	3705.70	1800.00	1096.10	3730.69	1695.39	1096	3736.78	1800.00	1109	3762.42	1800.00	1119.40	3790.12	1800.00	1135	3842.46	1800.00	914	3282.76	1800.00	920.50	3290.76	1800.00	927	3304.90	1800.00
E051-05e	50	769	2850.56	1652.77	776.30	2852.95	1454.16	773	2854.96	1405.77	797	2871.54	1800.00	794.30	2871.63	1772.49	794	2871.64	1800.00	799	2861.89	1800.00	803.50	2869.53	1800.00	801	2883.48	1800.00	648	2578.73	1800.00	666.60	2589.60	1800.00	681	2593.63	1800.00
E072-04f	71	515	1677.44	1800.00	523.00	1691.90	1800.00	530	1718.17	1800.00	512	1658.53	1800.00	515.80	1663.17	1800.00	519	1669.77	1800.00	525	1683.04	1800.00	527.10	1689.03	1800.00	529	1694.49	1800.00	457	1502.95	1800.00	458.70	1510.52	1800.00	463	1527.23	1800.00
E076-07s	75	1047	3807.47	1800.00	1054.20	3824.54	1800.00	1067	3846.14	1800.00	1135	3961.93	1800.00	1136.50	3978.49	1800.00	1147	3992.42	1800.00	991	3601.74	1800.00	1000.50	3612.46	1800.00	1008	3623.94	1800.00	893	3401.90	1800.00	904.50	3413.81	1800.00	909	3423.61	1800.00
E076-08s	75	1077	4063.90	1800.00	1080.30	4078.76	1800.00	1083	4110.26	1800.00	1088	4053.98	1800.00	1097.50	4077.25	1800.00	1105	4088.47	1800.00	1076	4011.51	1800.00	1090.20	4028.50	1800.00	1106	4045.37	1800.00	977	3747.38	1800.00	976.40	3757.08	1800.00	973	3784.62	1800.00
E076-10e	75	1077	4439.88	1800.00	1093.70	4493.34	1800.00	1104	4554.75	1800.00	1106	4413.87	1800.00	1119.00	4423.79	1800.00	1124	4431.89	1800.00	1114	4358.91	1800.00	1120.00	4376.71	1800.00	1121	4387.43	1800.00	978	4065.32	1800.00	977.90	4081.31	1800.00	992	4097.48	1800.00
E076-14s	75	1226	5260.99	1800.00	1221.60	5275.70	1800.00	1212	5295.91	1800.00	1124	4935.29	1800.00	1128.10	4950.94	1800.00	1131	4968.81	1800.00	1149	4992.21	1800.00	1140.70	5001.73	1800.00	1156	5026.86	1800.00	1058	4783.90	1800.00	1072.40	4802.40	1800.00	1079	4814.56	1800.00
E101-08e	100	1443	5007.84	3600.00	1440.10	5025.21	3600.00	1447	5062.55	3600.00	1401	4896.90	3600.00	1408.40	4916.57	3600.00	1414	4935.24	3600.00	1421	4900.91	3600.00	1415.80	4917.74	3600.00	1418	4940.02	3600.00	1161	4320.95	3600.00	1190.70	4350.32	3600.00	1195	4366.57	3600.00
E101-10c	100	1293	5087.44	3600.00	1296.60	5103.31	3600.00	1301	5112.75	3600.00	1364	5289.94	3600.00	1374.20	5301.24	3600.00	1380	5309.36	3600.00	1417	5428.18	3600.00	1427.50	5447.89	3600.00	1442	5466.83	3600.00	1245	4957.78	3600.00	1246.80	4965.33	3600.00	1249	4971.60	3600.00
E101-14s	100	1371	5707.51	3600.00	1382.60	5730.94	3600.00	1362	5747.86	3600.00	1426	5755.35	3600.00	1426.90	5764.61	3600.00	1428	5777.40	3600.00	1353	5572.86	3600.00	1356.70	5597.86	3600.00	1360	5616.86	3600.00	1263	5345.80	3600.00	1276.70	5360.88	3600.00	1266	5381.20	3600.00
E121-07c	120	2694	9244.83	3600.00	2705.50	9271.55	3600.00	2713	9293.95	3600.00	2765	9416.43	3600.00	2794.90	9489.49	3600.00	2817	9547.13	3600.00	2691	9221.00	3600.00	2738.40	9340.76	3600.00	2794	9496.10	3600.00	2460	8610.39	3600.00	2481.60	8669.06	3600.00	2522	8765.47	3600.00
E135-07f	134	2221	7939.87	3600.00	2254.80	8006.73	3600.00	2326	8188.48	3600.00	2114	7648.45	3600.00	2136.10	7677.42	3600.00	2155	7725.14	3600.00	2241	7941.28	3600.00	2263.30	7999.16	3600.00	2331	8158.86	3600.00	2139	7621.06	3600.00	2160.90	7699.28	3600.00	2183	7822.73	3600.00
E151-12b	150	1835	6580.92	3600.00	1855.20	6641.04	3600.00	1875	6671.39	3600.00	1871	6659.61	3600.00	1897.70	6707.46	3600.00	1915	6755.72	3600.00	1864	6602.77	3600.00	1887.60	6660.48	3600.00	1899	6689.92	3600.00	1571	5841.17	3600.00	1577.00	5871.20	3600.00	1595	5926.88	3600.00
E200-16b	199	2321	8473.10	3600.00	2351.20	8525.49	3600.00	2379	8578.87	3600.00	2338	8491.20	3600.00	2357.00	8535.04	3600.00	2379	8585.54	3600.00	2443	8737.09	3600.00	2471.20	8819.72	3600.00	2491	8901.10	3600.00	2054	7725.28	3600.00	2087.10	7804.72	3600.00	2114	7862.17	3600.00
E200-17b	199	2329	8521.11	3600.00	2359.60	8590.53	3600.00	2408	8700.11	3600.00	2328	8385.21	3600.00	2330.70	8455.16	3600.00	2338	8513.93	3600.00	2321	8408.03	3600.00	2344.50	8473.65	3600.00	2386	8592.83	3600.00	2031	7647.62	3600.00	2047.50	7707.84	3600.00	2069	7775.93	3600.00
E200-17c	199	2330	8483.48	3600.00	2360.90	8577.19	3600.00	2371	8649.72	3600.00	2429	8707.23	3600.00	2441.50	8737.43	3600.00	2463	8766.71	3600.00	2434	8711.43	3600.00	2459.90	8777.69	3600.00	2471	8833.05	3600.00	2054	7670.75	3600.00	2062.90	7739.52	3600.00	2076	7831.75	3600.00
E241-22k	240	1087	4082.64	3600.00	1088.10	4107.71	3600.00	1102	4137.92	3600.00	1098	4124.09	3600.00	1108.60	4143.69	3600.00	1120	4176.20	3600.00	1095	4094.53	3600.00	1097.30	4110.38	3600.00	1113	4133.56	3600.00	938	3682.77	3600.00	943.80	3694.18	3600.00	957	3718.01	3600.00
E253-27k	252	1386	5166.51	3600.00	1399.10	5201.41	3600.00	1424	5246.25	3600.00	1461	5318.45	3600.00	1475.90	5364.66	3600.00	1478	5390.04	3600.00	1514	5505.35	3600.00	1546.90	5579.58	3600.00	1565	5659.19	3600.00	1243	4736.63	3600.00	1251.40	4759.00	3600.00	1265	4783.69	3600.00
E256-14k	255	1719	5536.86	3600.00	1728.20	5574.06	3600.00	1750	5634.36	3600.00	1789	5720.10	3600.00	1815.20	5789.81	3600.00	1836	5856.01	3600.00	1649	5334.09	3600.00	1659.90	5368.40	3600.00	1670	5400.61	3600.00	1490	4912.34	3600.00	1502.10	4944.50	3600.00	1515	4985.40	3600.00

Table A.8: Complete results of the VNS for the G2L-SDVRP instances in Class 6.

Instance		VNS_{Best}			$VNS_{Average}$			VNS_{Worst}		
Name	n	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)	Sol_R	Sol_G	Time(s)
E016-03m	15	284	1152.93	7.46	284.00	1152.93	8.34	284	1152.93	9.70
E016-05m	15	308	1494.37	4.26	310.80	1494.37	6.04	312	1494.37	7.73
E021-04m	20	360	1577.67	94.48	363.00	1581.98	88.39	365	1585.04	89.16
E021-06m	20	427	1967.97	21.84	427.00	1967.97	25.79	427	1967.97	32.72
E022-04g	21	367	1751.55	156.02	367.00	1751.55	168.72	367	1751.55	186.22
E022-06m	21	479	2308.55	22.64	475.00	2312.01	46.27	471	2315.40	32.13
E023-03g	22	667	2460.34	671.70	692.80	2487.66	422.54	741	2539.19	330.95
E023-05s	22	655	2426.45	581.78	684.20	2476.38	438.70	690	2484.63	439.34
E026-08m	25	606	2845.57	38.84	606.00	2845.57	46.43	606	2845.57	60.93
E030-03g	29	637	2510.52	1609.72	647.50	2529.49	1415.99	650	2534.70	1256.13
E030-04s	29	637	2510.52	1613.00	647.50	2529.49	1414.98	650	2534.70	1246.10
E031-09h	30	585	2760.26	99.01	587.50	2770.39	66.73	588	2778.45	74.75
E033-03n	32	2424	8832.43	1409.83	2424.90	8834.57	1410.89	2433	8852.62	1488.63
E033-04g	32	1114	4626.17	1800.00	1113.30	4651.97	1800.00	1126	4689.35	1800.00
E033-05s	32	1114	4626.17	1800.00	1113.30	4651.97	1800.00	1126	4689.35	1800.00
E036-11h	35	689	3196.96	58.36	690.80	3199.64	105.55	695	3205.64	89.32
E041-14h	40	842	3954.48	117.47	847.10	3966.32	133.70	860	3979.05	135.22
E045-04f	44	949	3351.76	1800.00	955.90	3369.76	1800.00	967	3391.31	1800.00
E051-05e	50	689	2604.56	1800.00	686.00	2611.45	1800.00	677	2618.55	1800.00
E072-04f	71	438	1455.88	1800.00	443.90	1472.39	1800.00	449	1485.28	1800.00
E076-07s	75	901	3432.02	1800.00	920.20	3454.91	1800.00	933	3487.81	1800.00
E076-08s	75	910	3640.26	1800.00	929.20	3669.67	1800.00	940	3687.00	1800.00
E076-10e	75	943	3983.81	1800.00	955.80	4020.01	1800.00	967	4031.00	1800.00
E076-14s	75	1058	4760.28	1800.00	1053.30	4761.77	1800.00	1064	4765.66	1800.00
E101-08e	100	1181	4308.79	3600.00	1187.50	4333.72	3600.00	1205	4373.83	3600.00
E101-10c	100	1207	4862.55	3600.00	1215.70	4883.12	3600.00	1217	4903.49	3600.00
E101-14s	100	1196	5198.67	3600.00	1207.90	5217.11	3600.00	1213	5237.72	3600.00
E121-07c	120	2303	8211.17	3600.00	2344.30	8321.22	3600.00	2392	8407.97	3600.00
E135-07f	134	1970	7174.28	3600.00	1970.50	7237.47	3600.00	1985	7312.08	3600.00
E151-12b	150	1610	5909.77	3600.00	1617.50	5947.26	3600.00	1625	5972.23	3600.00
E200-16b	199	1977	7544.48	3600.00	2002.50	7586.15	3600.00	2026	7642.37	3600.00
E200-17b	199	1989	7526.24	3600.00	2002.30	7573.83	3600.00	2025	7645.41	3600.00
E200-17c	199	1957	7471.74	3600.00	1983.20	7531.18	3600.00	2012	7583.38	3600.00
E241-22k	240	862	3489.76	3600.00	876.50	3520.40	3600.00	878	3538.16	3600.00
E253-27k	252	1205	4630.11	3600.00	1212.20	4650.32	3600.00	1221	4705.81	3600.00
E256-14k	255	1400	4673.65	3600.00	1410.10	4697.02	3600.00	1412	4712.29	3600.00

Table A.9: Improvements obtained with the VNS compared to its initial solution for the G2L-SDVRP instances of Classes 2 to 5.

Instances	Class 2						Class 3						Class 4						Class 5					
	Sol_I	Sol_F	$Gap(\%)$	T_I	T_F	RT	Sol_I	Sol_F	$Gap(\%)$	T_I	T_F	RT	Sol_I	Sol_F	$Gap(\%)$	T_I	T_F	RT	Sol_I	Sol_F	$Gap(\%)$	T_I	T_F	RT
E016-03m	2158.28	1329.64	-38.39	0.40	47.43	0.84	2103.28	1345.90	-36.01	1.57	98.66	1.59	1764.70	1195.44	-32.26	0.32	20.72	1.54	1419.54	1152.93	-18.78	0.03	26.85	0.11
E016-05m	2262.93	1558.24	-31.14	0.06	5.45	1.10	2258.18	1558.88	-30.97	0.15	13.71	1.09	2200.37	1494.37	-32.09	0.09	8.97	1.00	2195.38	1494.37	-31.93	0.01	4.57	0.22
E021-04m	2454.42	1683.62	-31.40	0.01	23.27	0.04	2748.99	1638.97	-40.38	0.48	44.59	1.08	2369.38	1605.27	-32.25	0.01	45.35	0.02	2404.80	1577.67	-34.39	0.02	20.63	0.10
E021-06m	3319.65	2022.97	-39.06	0.18	12.64	1.42	3114.19	1966.85	-36.84	0.07	19.86	0.35	3112.89	1995.89	-35.88	0.58	34.15	1.70	3112.52	2002.84	-35.65	1.07	27.99	3.82
E022-04g	2395.89	1811.93	-24.37	0.03	47.94	0.06	3400.79	1769.66	-47.96	0.55	44.60	1.23	3511.71	1825.55	-48.02	3.71	87.36	4.25	2389.17	1751.55	-26.69	0.00	20.79	0.00
E022-06m	3331.03	2326.14	-30.17	0.01	18.81	0.05	2877.76	2338.15	-18.75	0.03	27.63	0.11	4283.69	2308.89	-46.10	0.70	167.82	0.42	3531.43	2308.77	-34.62	0.01	23.25	0.04
E023-03g	3134.08	2547.63	-18.71	0.06	49.22	0.12	3115.91	2502.23	-19.70	0.05	154.11	0.03	3296.59	2498.22	-24.22	0.02	287.30	0.01	3121.34	2437.96	-21.89	0.11	162.63	0.07
E023-05s	3309.04	2464.52	-25.52	0.04	39.66	0.10	3687.80	2564.96	-30.45	0.11	72.81	0.15	3377.03	2494.76	-26.13	0.11	121.42	0.09	3799.18	2393.99	-36.99	0.24	166.86	0.14
E026-08m	4346.82	2879.20	-33.76	0.17	37.28	0.46	3747.62	2885.36	-23.01	0.09	135.23	0.07	4012.02	2886.15	-28.06	0.11	116.86	0.09	4488.15	2797.82	-37.66	0.57	65.10	0.88
E030-03g	3940.13	2663.79	-32.39	0.05	160.06	0.03	3400.22	2503.73	-26.37	0.09	282.77	0.03	4068.62	2657.68	-34.68	0.12	417.97	0.03	3237.18	2625.00	-18.91	0.19	673.49	0.03
E030-04s	4038.35	2734.84	-32.28	0.13	156.12	0.08	3993.13	2674.84	-33.01	0.06	462.84	0.01	3633.26	2894.96	-20.32	0.10	1131.61	0.01	3614.88	2483.86	-31.29	0.43	1126.28	0.04
E031-09h	4551.56	2876.08	-36.81	1.52	78.97	1.92	4275.12	2794.28	-34.64	0.25	34.46	0.73	4697.85	2844.75	-39.45	1.21	332.41	0.36	4815.73	2762.39	-42.64	1.47	146.49	1.00
E033-03n	13395.60	9528.91	-28.87	0.04	138.14	0.03	14756.90	9101.12	-38.33	0.03	332.73	0.01	13930.70	9269.25	-33.46	0.15	1800.00	0.01	14386.70	8787.90	-38.92	0.30	1800.00	0.02
E033-04g	5839.95	4718.85	-19.20	0.05	288.97	0.02	6172.88	4519.54	-26.78	0.09	1661.64	0.01	5413.66	4361.11	-19.44	0.23	1447.64	0.02	5595.79	4204.27	-24.87	0.65	1800.00	0.04
E033-05s	5877.18	4548.26	-22.61	0.04	612.03	0.01	7227.10	4823.86	-33.25	0.70	446.09	0.16	5935.82	4921.60	-17.09	0.17	1800.00	0.01	5922.88	4912.87	-17.05	0.41	1800.00	0.02
E036-11h	5494.30	3229.32	-41.22	0.29	52.49	0.55	5051.30	3230.41	-36.05	0.09	119.36	0.08	5342.54	3230.45	-39.53	0.13	236.74	0.05	5182.83	3230.17	-37.68	0.17	66.92	0.25
E041-14h	6585.39	3981.83	-39.54	0.33	72.16	0.46	6706.80	3972.59	-40.77	0.04	84.22	0.05	6490.61	3976.00	-38.74	0.13	103.65	0.13	6432.78	3972.50	-38.25	0.07	114.77	0.06
E045-04f	5415.55	3620.90	-33.14	0.08	1722.59	0.00	4425.75	3705.70	-16.27	0.12	1800.00	0.01	5148.74	3762.42	-26.93	0.23	1800.00	0.01	5831.13	3282.76	-43.70	1.09	1800.00	0.06
E051-05e	4242.97	2850.56	-32.82	0.28	1652.77	0.02	4051.51	2871.54	-29.12	0.08	1800.00	0.00	4279.99	2861.89	-33.13	0.69	1800.00	0.04	4139.91	2578.73	-37.71	0.93	1800.00	0.05
E072-04f	2682.84	1677.44	-37.48	0.85	1800.00	0.05	2533.43	1658.53	-34.53	1.57	1800.00	0.09	2217.19	1683.04	-24.09	0.41	1800.00	0.02	1971.55	1502.95	-23.77	1.03	1800.00	0.06
E076-07s	7061.91	3807.47	-46.08	1.34	1800.00	0.07	5556.56	3961.93	-28.70	0.18	1800.00	0.01	5195.56	3601.74	-30.68	0.36	1800.00	0.02	5536.44	3401.90	-38.55	0.63	1800.00	0.04
E076-08s	6919.94	4063.90	-41.27	0.92	1800.00	0.05	6078.35	4053.98	-33.30	0.99	1800.00	0.06	5749.04	4011.51	-30.22	1.47	1800.00	0.08	5879.32	3747.38	-36.26	1.05	1800.00	0.06
E076-10e	7909.37	4439.88	-43.87	1.53	1800.00	0.09	6680.04	4413.87	-33.92	0.74	1800.00	0.04	7534.46	4358.91	-42.15	5.72	1800.00	0.32	5935.79	4065.32	-31.51	0.60	1800.00	0.03
E076-14s	9181.62	5260.99	-42.70	0.68	1800.00	0.04	7386.20	4935.29	-33.18	0.80	1800.00	0.04	7783.84	4992.21	-35.86	1.55	1800.00	0.09	7246.75	4783.90	-33.99	0.41	1800.00	0.02
E101-08e	7647.24	5007.84	-34.51	0.58	3600.00	0.02	7894.50	4896.90	-37.97	3.32	3600.00	0.09	7310.77	4900.91	-32.96	6.48	3600.00	0.18	6866.28	4320.95	-37.07	1.75	3600.00	0.05
E101-10c	8665.30	5087.44	-41.29	1.82	3600.00	0.05	7246.86	5289.94	-27.00	2.57	3600.00	0.07	8770.74	5428.18	-38.11	11.90	3600.00	0.33	6347.59	4957.78	-21.90	1.19	3600.00	0.03
E101-14s	8578.94	5707.51	-33.47	1.22	3600.00	0.03	8429.60	5755.35	-31.72	1.46	3600.00	0.04	8365.80	5572.86	-33.39	0.40	3600.00	0.01	8441.84	5345.80	-36.67	1.42	3600.00	0.04
E121-07c	12858.00	9244.83	-28.10	2.57	3600.00	0.07	11071.60	9416.43	-14.95	0.65	3600.00	0.02	11203.30	9221.00	-17.69	3.06	3600.00	0.09	9883.25	8610.39	-12.88	3.54	3600.00	0.10
E135-07f	14246.10	7939.87	-44.27	5.53	3600.00	0.15	10947.20	7648.45	-30.13	3.44	3600.00	0.10	12490.00	7941.28	-36.42	10.49	3600.00	0.29	9657.04	7621.06	-21.08	2.16	3600.00	0.06
E151-12b	11414.30	6580.92	-42.34	2.97	3600.00	0.08	12755.80	6659.61	-47.79	15.24	3600.00	0.42	11219.90	6602.77	-41.15	15.68	3600.00	0.44	10154.50	5841.17	-42.48	3.13	3600.00	0.09
E200-16b	13847.70	8473.10	-38.81	7.34	3600.00	0.20	14776.50	8491.20	-42.54	16.26	3600.00	0.45	14222.80	8737.09	-38.57	26.53	3600.00	0.74	12118.60	7725.28	-36.25	3.79	3600.00	0.11
E200-17b	14038.70	8521.11	-39.30	6.29	3600.00	0.17	13430.40	8385.21	-37.57	7.19	3600.00	0.20	14450.80	8408.03	-41.82	36.72	3600.00	1.02	12567.00	7647.62	-39.15	6.57	3600.00	0.18
E200-17c	17516.30	8483.48	-51.57	10.26	3600.00	0.29	12761.00	8707.23	-31.77	9.48	3600.00	0.26	13986.90	8711.43	-37.72	29.09	3600.00	0.81	11328.20	7670.75	-32.29	3.13	3600.00	0.09
E241-22k	7225.88	4082.64	-43.50	11.84	3600.00	0.33	6248.06	4124.09	-33.99	11.89	3600.00	0.33	6772.61	4094.53	-39.54	49.26	3600.00	1.37	5252.76	3682.77	-29.89	4.82	3600.00	0.13
E253-27k	10361.90	5166.51	-50.14	21.39	3600.00	0.59	7894.61	5318.45	-32.63	21.58	3600.00	0.60	11584.20	5505.35	-52.48	118.08	3600.00	3.28	5961.48	4736.63	-20.55	7.47	3600.00	0.21
E256-14k	7487.28	5536.86	-26.05	18.40	3600.00	0.51	7545.22	5720.10	-24.19	23.16	3600.00	0.64	6249.99	5334.09	-14.65	4.57	3600.00	0.13	5907.36	4912.34	-16.84	19.53	3600.00	0.54

Table A.10: Improvements obtained with the VNS compared to its initial solution for the G2L-SDVRP instances of Class 6.

Instances	VNS solution					
	Sol_I	Sol_F	$Gap(\%)$	T_I	T_F	RT
E016-03m	1399.16	1152.93	-17.60	0.00	7.46	0.00
E016-05m	2239.99	1494.37	-33.29	0.02	4.26	0.47
E021-04m	2543.54	1577.67	-37.97	0.07	94.48	0.07
E021-06m	3106.67	1967.97	-36.65	0.10	21.84	0.46
E022-04g	2470.41	1751.55	-29.10	0.11	156.02	0.07
E022-06m	3765.33	2308.55	-38.69	0.02	22.64	0.09
E023-03g	3226.23	2460.34	-23.74	0.34	671.70	0.05
E023-05s	3226.23	2426.45	-24.79	0.34	581.78	0.06
E026-08m	4526.82	2845.57	-37.14	0.45	38.84	1.16
E030-03g	3472.64	2510.52	-27.71	0.36	1609.72	0.02
E030-04s	3472.64	2510.52	-27.71	0.36	1613.00	0.02
E031-09h	4423.70	2760.26	-37.60	0.01	99.01	0.01
E033-03n	13816.30	8832.43	-36.07	0.45	1409.83	0.03
E033-04g	6454.57	4626.17	-28.33	0.66	1800.00	0.04
E033-05s	6454.57	4626.17	-28.33	0.66	1800.00	0.04
E036-11h	5336.44	3196.96	-40.09	0.01	58.36	0.02
E041-14h	5850.39	3954.48	-32.41	0.00	117.47	0.00
E045-04f	4501.05	3351.76	-25.53	1.40	1800.00	0.08
E051-05e	4555.68	2604.56	-42.83	0.81	1800.00	0.05
E072-04f	2051.81	1455.88	-29.04	1.01	1800.00	0.06
E076-07s	5460.04	3432.02	-37.14	1.84	1800.00	0.10
E076-08s	5917.82	3640.26	-38.49	2.21	1800.00	0.12
E076-10e	6722.73	3983.81	-40.74	1.54	1800.00	0.09
E076-14s	7519.89	4760.28	-36.70	0.56	1800.00	0.03
E101-08e	6939.74	4308.79	-37.91	3.43	3600.00	0.10
E101-10c	6393.05	4862.55	-23.94	1.77	3600.00	0.05
E101-14s	8555.26	5198.67	-39.23	2.64	3600.00	0.07
E121-07c	9512.14	8211.17	-13.68	5.02	3600.00	0.14
E135-07f	10028.70	7174.28	-28.46	6.48	3600.00	0.18
E151-12b	8889.45	5909.77	-33.52	4.97	3600.00	0.14
E200-16b	11691.80	7544.48	-35.47	7.09	3600.00	0.20
E200-17b	11691.80	7526.24	-35.63	6.62	3600.00	0.18
E200-17c	11597.70	7471.74	-35.58	6.01	3600.00	0.17
E241-22k	4931.31	3489.76	-29.23	5.65	3600.00	0.16
E253-27k	6203.04	4630.11	-25.36	11.05	3600.00	0.31
E256-14k	5467.73	4673.65	-14.52	22.19	3600.00	0.62
Average			-31.67	2.67	1830.73	0.15

Table A.11: Results of the BC method of Ferreira et al. (2021) (with the normal patterns) for the G2L-SDVRP instances in Classes 1 to 5.

Instances				Banch-and-cut solution								
Name	Class	n	R	K_{min}	K	VH	CS	Sol_R	Sol_G	$Time_T$	$Time_P$	Cut_P
E016-03m	1	15	15	3	3	3	0	273	1277.63	6.28	0.00	0
	2	15	24	3	3	3	1	282	1329.64	236.56	8.49	66
	3	15	31	3	3	3	2	291	1345.90	824.73	101.98	115
	4	15	37	3	4	4	0	288	1195.44	93.17	75.50	13
	5	15	45	3	4	4	0	284	1152.93	3.78	0.00	0
E016-05m	1	15	15	5	5	5	0	340	1561.18	3.78	0.00	0
	2	15	25	5	5	5	1	330	1558.24	151.82	0.16	11
	3	15	31	5	5	5	3	333	1558.88	214.02	0.70	17
	4	15	40	5	5	5	1	308	1494.37	71.16	0.07	0
	5	15	48	5	5	5	1	312	1494.37	46.52	0.04	0
E021-04m	1	20	20	4	4	4	0	372	1642.42	50.62	0.00	0
	2	20	29	4	5	5	1	402	1683.62	3556.15	1.83	143
	3	20	46	4	5	5	0	391	1638.97	3592.58	34.09	10
	4	20	44	4	5	5	1	380	1605.27	653.54	0.04	0
	5	20	49	4	5	5	0	360	1577.67	338.41	0.07	0
E021-06m	1	20	20	6	6	6	0	447	2025.85	30.65	0.00	0
	2	20	32	6	6	6	1	443	2022.97	857.03	0.02	0
	3	20	43	6	6	6	2	427	1966.85	629.00	0.07	1
	4	20	50	6	6	6	1	436	1995.89	2372.69	0.16	0
	5	20	62	6	6	6	1	436	2002.84	3598.53	0.00	0
E022-04g	1	21	21	4	4	4	0	367	1751.55	4.98	0.00	0
	2	21	31	4	4	4	1	382	1811.93	159.58	1.37	22
	3	21	37	4	4	4	0	373	1769.66	64.41	0.08	2
	4	21	41	4	4	4	1	377	1825.55	478.57	39.55	305
	5	21	57	4	5	4	0	367	1751.55	48.17	0.03	0
E022-06m	1	21	21	6	6	6	0	492	2341.64	55.50	0.00	0
	2	21	33	6	6	6	1	473	2326.14	415.44	0.12	5
	3	21	40	6	6	6	1	499	2338.15	3580.31	4.19	13
	4	21	57	6	6	6	2	479	2308.78	1772.29	0.27	0
	5	21	56	6	6	6	2	479	2308.77	2928.73	0.11	0
E023-03g	1	22	22	3	3	3	0	564	2298.92	29.41	0.00	0
	2	22	32	4	5	5	0	725	2553.81	3597.16	2.74	108
	3	22	41	4	5	5	0	732	2518.70	3596.29	111.07	79
	4	22	51	4	5	5	0	708	2515.22	3569.34	262.01	6
	5	22	55	3	6	4	1	650	2444.04	3589.27	605.40	5
E023-05s	1	22	22	3	5	3	0	564	2298.92	76.26	0.00	0
	2	22	29	4	5	4	2	681	2464.52	3582.36	4.23	188
	3	22	42	4	5	5	1	750	2564.96	3575.98	0.05	0
	4	22	48	4	5	5	2	699	2494.53	3590.68	974.88	41
	5	22	52	3	6	4	0	632	2413.90	3599.17	989.06	9
E026-08m	1	25	25	8	8	8	0	610	2897.58	163.52	0.00	0
	2	25	40	8	8	8	2	613	2879.20	3597.51	0.03	0
	3	25	61	8	8	8	1	619	2917.75	3591.87	0.25	3
	4	25	63	8	8	8	4	621	2880.88	3522.69	0.28	0
	5	25	91	8	8	8	4	594	2790.43	3597.66	0.08	0
E030-03g	1	29	29	3	3	3	0	549	2523.88	3299.82	0.00	0
	2	29	43	5	6	6	0	709	2672.04	3299.87	1.48	47
	3	29	49	4	6	5	2	622	2512.13	3299.97	77.37	49
	4	29	72	6	7	6	1	703	2657.68	3368.04	1003.50	26
	5	29	86	5		6	2	689	2632.92	3490.29	842.50	7
E033-03n	1	32	32	3	3	3	0	2034	8145.22	442.47	0.00	0
	2	32	44	5	7	7	1	3067	10515.70	3299.85	9.41	506
	3	32	56	5	7	7	1	2576	9218.05	3299.95	109.03	115
	4	32	78	6	7	6	2	2623	9396.29	3276.52	760.22	71
	5	32	102	5	8	5	4	2572	9258.79	3477.68	2533.93	20
E036-11h	1	35	35	11	11	11	0	708	3274.33	3299.96	0.00	0
	2	35	56	11	11	11	1	707	3268.83	3299.92	0.01	0
	3	35	74	11	11	11	0	711	3291.48	3299.91	0.07	1
	4	35	93	11	11	11	3	722	3272.57	3299.95	0.24	1
	5	35	114	11	11	11	0	708	3274.33	3299.90	0.00	0

Table A.12: Results of the BC method of Ferreira et al. (2021) (with the normal patterns) for the G2L-SDVRP instances in Class 6.

Instances			Branch-and-cut solution								
Name	n	R	K_{min}	K	VH	CS	Sol_R	Sol_G	$Time_T$	$Time_P$	Cut_P
E016-03m	15	48	3	4	4	0	284	1152.93	4.40	0.00	0
E016-05m	15	48	5	5	5	1	308	1494.37	66.16	0.00	0
E021-04m	20	66	4	5	5	0	360	1577.67	818.81	0.21	0
E021-06m	20	66	6	6	6	2	427	1967.97	3598.25	0.00	0
E022-04g	21	68	4	5	4	0	367	1751.55	158.19	0.35	0
E022-06m	21	68	6	6	6	2	473	2311.50	3596.45	0.00	0
E023-03g	22	70	4	6	4	2	653	2414.24	3581.86	2.24	0
E023-05s	22	70	4	6	4	2	653	2414.24	3581.81	3.55	0
E026-08m	25	79	8	8	8	3	598	2828.42	3597.31	0.30	0
E030-03g	29	91	5	7	7	2	662	2560.66	3596.88	245.60	1
E033-03n	32	99	5	8	6	3	2439	8981.47	3572.22	1458.16	14
E036-11h	35	109	11	11	11	5	707	3250.60	3299.97	0.00	0