The impact of loading restrictions on the two-echelon location routing problem

Vinícius Martins Santos Gandra[a,b], Hatice Çalık[a], Tony Wauters[a], Túlio A. M. Toffolo[b], Marco Antonio Moreira de Carvalho[b], Greet Vanden Berghe[a]

[a] *KU Leuven, Dept. of Computer Science; Leuven.AI, B-3000 Leuven, Belgium*
[b] *Department of Computing, Federal University of Ouro Preto, Brazil*

Abstract

The two-echelon location routing problem (2E-LRP) arises in freight distribution when goods available at different origins are delivered to their respective destinations via intermediate facilities. The literature concerning the 2E-LRP considers freight capacities of vehicles to be scalars, while customer demands are additive volumes of individual items. However, ignoring the real dimensions of items and vehicles can lead to infeasible load plans in practice. Further investigation is thus required to study the impact of realistic loading restrictions on the 2E-LRP, algorithms for the problem and the quality of the solutions produced by those algorithms. This paper introduces a generalized 2E-LRP with two-dimensional loading restrictions (2E-LRP2L). To investigate how exactly one should handle these restrictions we introduce a heuristic optimization method combined with different loading strategies and evaluate their performance on instances derived from real-world data. Given that we are introducing a new problem, the quality of our heuristic is assessed by comparing it against state-of-the-art 2E-LRP methods on benchmark instances. The results indicate that the proposed method is highly competitive, finding most best-known solutions as well as providing some new ones.

Keywords: Two echelon, Vehicle Routing, Location Routing, 2D-packing.

1. Introduction

This paper addresses a real-world problem emerging from a logistics company which supplies its customers with large material handling equipment as well as industrial and agricultural machines. The company chooses from the depots (platforms) of several external providers to satisfy the demand of its customers. When a platform is utilized, a fixed rental or contract cost must be paid. Using

Email addresses: vinicius.gandramartinssantos@kuleuven.be (Vinícius Martins Santos Gandra), hatice.calik@kuleuven.be (Hatice Çalık), tony.wauters@kuleuven.be (Tony Wauters), tulio@toffolo.com.br (Túlio A. M. Toffolo), mamc@ufop.edu.br (Marco Antonio Moreira de Carvalho), greet.vanden.berghe@cs.kuleuven.be (Greet Vanden Berghe)

capacitated vehicles the items dispatched from these platforms are then delivered to parking areas or sorting facilities (satellites), which must also be selected from multiple options. As with platforms, using a satellite incurs a fixed rental cost. The distribution of items to customers is then performed from these intermediate facilities. Both the platforms and satellites have a limited service capacity and thus the total demand allocated to each facility should not exceed its capacity.

In order to immediately ground the problem in physical reality, Figure 1 illustrates a toy network involving four candidate platforms, six candidate satellites and 15 customers. The first distribution level (echelon) comprises of the two open platforms (p_2 and p_4) and the routes connecting these platforms to the satellites (represented with the thicker lines). The second echelon comprises of three open satellites: s_3, s_4 and s_5. Each customer (1-15) is served by a single vehicle whose origin is one of these satellites. The routes connecting the smaller second-echelon vehicles with customers are represented with thinner lines. An example of a load plan, the exact position of items in the vehicle's two-dimensional loading surface, is provided for the vehicle during route $s_4 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow s_4$. Items are loaded from the back of the vehicle (rear loading) at satellite s_4 before the trip begins. During the trip, each item will be unloaded from the rear. In this particular load plan, the items belonging to customers 9 and 10 are positioned behind those belonging to customers 8 and 7, respectively. Thanks to this positioning, the unloading of items at customer locations is possible without having to move any other items out of the way.

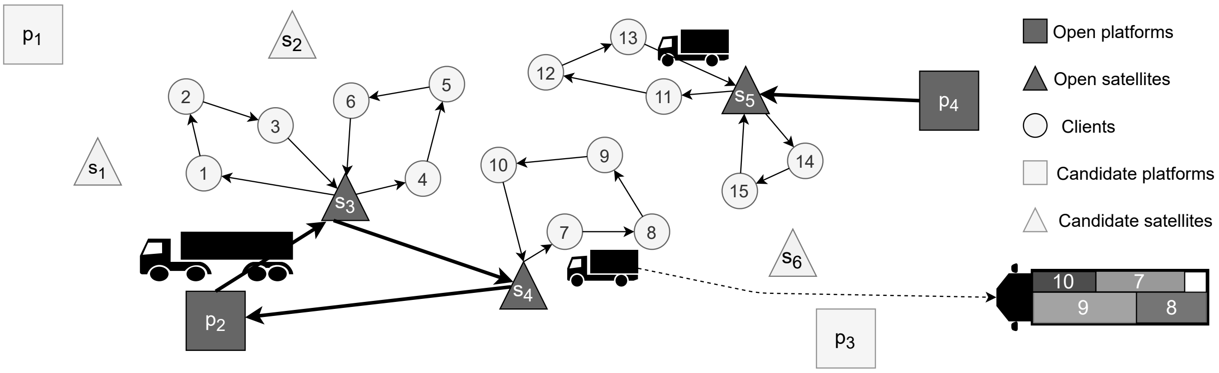


Figure 1: Two-echelon routes with load plan.

The literature on multi-echelon location-routing problems dates back to the early study by Jacobsen and Madsen (1980), while Boccia et al. (2010) first described the two-echelon location routing problem (2E-LRP) where both platforms and satellites must be selected from multiple candidates. The 2E-LRP, the most closely related problem to the presented scenario, considers only one dimension (either area or volume) when loading items into vehicles. The logistics company that introduced us to the problem we address in this paper faces a similar situation insofar as they utilize the area of a rectangular-shaped projection of the items (on the ground surface) in their calculations when assigning items to facilities and vehicles. However, this approximation of item dimensions (length and width) does not always provide accurate results when loading said items inside the first- and second-echelon vehicles. As a result of ignoring the items' two-dimensional footprint, the company generates load plans with items which will never fit together. When attempting to implement these infeasible load plans, the company may be forced to outsource the delivery of some items to third-party vehicles. Given that this can prove very expensive, in this paper we take into account realistic two-dimensional loading constraints when assigning items to vehicles (which are also characterized by their length and width).

The choice of two-dimensional loading constraints is justified by the real-world context which motivated this research. Given that items are extremely large and heavy machinery products, they cannot overlap with one another or be stacked without causing significant damage or deformation. Furthermore, items have wheels with limited maneuvering capacity and are either driven into vehicles or (un)loaded in a specific direction using forklifts. Either way, this (un)loading procedure means that rotating items is impossible.Therefore, it is perfectly realistic to ignore the height of these items and consider only their two-dimensional projection (bottom surface).

The problem thus becomes a generalization of the 2E-LRP which we term the *two-echelon location-routing problem with two-dimensional loading constraints* (2E-LRP2L). In the particular application considered in this paper, the total demand assigned to a satellite may be larger than the capacity of first-echelon vehicles. Therefore, it is permitted to split deliveries to satellites across multiple vehicles in the first echelon. Moreover, besides the standard two-dimensional loading constraints, unloading and re-loading an item at the same location or movements inside the vehicle are considered highly undesirable. This is due to the fact that the space required for such movements is limited. Hence when these movements and item relocations take place, a *sequential loading penalty* is incurred.

The 2E-LRP2L is extremely challenging as it integrates $\mathcal{NP}$-complete loading decisions (Fowler et al., 1981) into the 2E-LRP which, on its own, is already an $\mathcal{NP}$-hard problem. Given that the 2E-LRP is difficult to solve with exact methods, one can logically assume that such methods will also be impractical when it comes to solving medium- and large-scale 2E-LRP2L instances within a reasonable amount of time. Therefore, this paper introduces a heuristic approach to generate solutions for the 2E-LRP2L.

Besides introducing a heuristic for the 2E-LRP2L, this paper will also study the effects of incorporating realistic loading constraints into the 2E-LRP and the impact of items' loading order on solution quality. To quantify these effects, four loading strategies are developed for the proposed heuristic. These different strategies should enable us to arrive at a better understanding of the problem's unique features. A set of instances is derived from real customer locations and requests, as well as real vehicle and item sizes.

The remainder of the paper is organized as follows. Section 2 presents a literature review concerning related problems and indicates relevant elements for the present study. A detailed description of the problem and notation is provided in Section 3, while the proposed method for the 2E-LRP2L is introduced in Section 4. Computational experiments and a study on the impact of loading constraints are conducted in Section 5. Section 6 makes a few minor modifications to the proposed algorithm so as to make it applicable to 2E-LRP instances. Finally, Section 7 concludes this paper and outlines some future research directions.

2. Related work

This section initially focuses on detailing the main contributions made in relation to the 2E-LRP, which are recent and few in number, while also providing an overview of some closely related problems in Section 2.1. Later, in Section 2.2, the most relevant studies on the vehicle routing problem with two-dimensional loading constraints (2L-CVRP) are discussed. The 2E-LRP considers location and routing decisions in two echelons and is a generalization of the two-echelon vehicle routing problem (2E-VRP) first introduced by Perboli et al. (2011). Location and routing have often been studied in an intertwined manner since assessing such decisions separately may result in excessive overall costs (Salhi and Rand, 1989; Salhi and Nagy, 1999).

The studies reviewed in this section are limited to those which had a direct impact on this paper. Table 1 provides an overview of those studies and details the problem(s) each addresses along with the decisions involved when doing so. The general observations made in the following subsections are supported by the papers from this table. We refer readers interested in more extensive reviews to the following surveys. Cuda et al. (2015) and Drexl and Schneider (2015) review two echelon routing problems that include location routing, vehicle routing, as well as truck and trailer problems. Nagy and Salhi (2007), Prodhon and Prins (2014) and Schneider and Drexl (2017) provide complementary reviews on location-routing problems. Laporte (2009), Kumar and Panneerselvam (2012) and Vidal et al. (2019) discuss the vehicle routing problem and its variants. Finally, Bortfeldt and Wäscher (2013), Iori and Martello (2010) and Pollaris et al. (2015) present a broad overview concerning container loading problems and vehicle routing problems with two- and three-dimensional loading constraints.

2.1. Selected studies concerning the 2E-LRP

Boccia et al. (2010) were the first to formalize the 2E-LRP. They decomposed the problem into two location routing problems (LRPs), one per echelon. A bottom-up approach was employed whereby the solution for the first echelon was generated and heuristically improved based on the solution of the second. Each echelon was decomposed into a capacitated facility location problem (the location phase) and a multi-depot vehicle routing problem (the routing phase). This decomposition approach facilitates the development of dedicated location and routing heuristics to solve the problem and is the most common approach adopted in the literature.

Assessing and comparing the quality of methods proposed for the 2E-LRP is possible thanks to Boccia et al. (2010) and Nguyen et al. (2012b), who generated benchmark instances and made them publicly available. Nguyen et al. (2012b) proposed instances with a single platform, a special

Table 1: Overview of 2E-LRP and 2l-CVRP approaches and related literature.

Reference	Problem(s) addressed	Method	Two-echelon	Location	Routing	Multi platform	Split delivery	Packing
Boccia et al. (2010)	2E-LRP	H	✓	✓	✓	✓	-	-
Boccia et al. (2011)	2E-LRP	E	✓	✓	✓	✓	-	-
Nguyen et al. (2012a)	2E-LRP and 2E-LRPSD	H	✓	✓	✓	✓	-	-
Nguyen et al. (2012b)	2E-LRPSD	H	✓	✓	✓	-	-	-
Contardo et al. (2012)	2E-LRP and 2E-LRPSD	E and H	✓	✓	✓	✓	-	-
Schwengerer et al. (2012)	2E-LRP and 2E-LRPSD	H	✓	✓	✓	✓	-	-
Breunig et al. (2016)	2E-VRP and 2E-LRPSD	H	✓	✓	✓	-	-	-
Pichka et al. (2018)	2E-OLRPSD and 2E-LRPSD	E and H	✓	✓	✓	-	-	-
Iori et al. (2007)	2L-CVRP	E	-	-	✓	-	-	✓
Gendreau et al. (2008)	2L-CVRP	H	-	-	✓	-	-	✓
Zachariadis et al. (2009)	2L-CVRP	H	-	-	✓	-	-	✓
Fuellerer et al. (2009)	2L-CVRP	H	-	-	✓	-	-	✓
Leung et al. (2010)	2L-CVRP	H	-	-	✓	-	-	✓
Leung et al. (2011)	2L-CVRP	H	-	-	✓	-	-	✓
Zachariadis et al. (2013)	2L-CVRP	H	-	-	✓	-	-	✓
Wei et al. (2015)	2L-CVRP	H	-	-	✓	-	–	✓
Wei et al. (2018)	2L-CVRP	H	-	-	✓	-	-	✓
This paper	2E-LRP2L and 2E-LRP(SD)	H	✓	✓	✓	✓	✓	✓

E: exact approach;
H: heuristic;
2E-OLRPSD: 2E-LRPSD with open routes (vehicles do not need to return to depots).

case referred to as the 2E-LRPSD, which has no location decisions in the first echelon. All relevant papers for the 2E-LRP have since employed these instances to assess the performance of their methods.

The two current state-of-the-art methods for the 2E-LRP were introduced by Contardo et al. (2012) and Schwengerer et al. (2012). Contardo et al. (2012) proposed a two-index vehicle-flow formulation that serves as the base for a branch-and-cut (B&C) algorithm. This exact B&C method is able to solve instances with up to 50 customers and 10 satellites to optimality. Moreover, Contardo et al. (2012) proposed an adaptive large neighborhood search (ALNS) heuristic. Their heuristic decomposes the 2E-LRP into two LRPs and is complemented by five local search operators which improve the second-echelon LRP. This ALNS method outperformed previous heuristics proposed by Nguyen et al. (2012a) and Nguyen et al. (2012b). The second state-of-the-art method, Schwengerer et al. (2012), addressed the 2E-LRP with an extension of the variable neighborhood search (VNS) originally introduced for the LRP by Pirkwieser and Raidl (2010). Their method was able to generate new best solutions, but no significant difference concerning solution quality is evident when comparing the VNS with the ALNS of Contardo et al. (2012).

There are a few observations worth stressing concerning the research related to the 2E-LRP and the 2E-LRPSD documented in Table 1. The table indicates that high-quality solutions for medium- and large-scale instances of the 2E-LRP are most often obtained with heuristics. Fast, dedicated approaches are typically employed for the first echelon given its small size relative to the second echelon. However, it is worth noting that Boccia et al. (2010) and Contardo et al. (2012) recursively address both echelons using a single method.

2.2. Selected studies concerning the 2L-CVRP

When it comes to loading restrictions, the most closely related problem to the 2E-LRP2L is the 2L-CVRP. Although a generalization of the 2L-CVRP which considers a VRP with three-dimensional loading constraints (3L-CVRP) was introduced by Gendreau et al. (2006), the first dedicated method for the 2L-CVRP was provided by Iori et al. (2007), who developed an exact approach. Wei et al. (2018) later introduced a simulated annealing (SA) framework combined with efficient data structures and a heuristic based on free spaces (unpacked regions) for the loading of items. Wei et al. (2018) outperform all previous approaches for the four variants of the problem: with/without sequential loading while allowing/prohibiting item rotation.

Various heuristic approaches with satisfactory results for the CVRP have also been extended to address the 2L-CVRP. Studies concerning the 2L-CVRP documented in Table 1 range from the paper which first introduced the problem (Iori et al., 2007) through to the state of the art (Wei et al., 2018). All these papers employ a two-stage approach, where the routing is treated as the main problem and for each generated or candidate route a packing heuristic is called to solve the loading subproblem. Packing methods are generally constructive and the load plans computed are stored together with the routes in a memory structure to avoid re-computation. For each generated route, the memory structure is first checked for the given route's load plan and the packing method is only called if there is no load plan associated with the generated route in memory.

3. Problem description

The 2E-LRP2L can be formally described with a weighted undirected graph $G = (V, E)$. $V = P \cup S \cup C$ is the vertex set where the three disjoint sets P, S and C correspond to the sets of platforms, satellites and customers, respectively. $E = E^1 \cup E^2$ is the set of all edges, where $E^1 = \{\{i, j\} : i, j \in P \cup S, i \neq j, \{i, j\} \notin P \times P\}$ corresponds to the edges in the first echelon while $E^2 = \{\{i, j\} : i, j \in S \cup C, i \neq j, \{i, j\} \notin S \times S\}$ corresponds to the edges in the second echelon. Each $i \in P \cup S$ is associated with a capacity Q_i (given in square meters) and a fixed cost H_i incurred upon usage. Each edge $\{i, j\} \in E$ is associated with a routing cost $e_{ij} > 0$ proportional to its travel time.

Let M be the set of items requested by all customers and $M_c \subset M$ be the set of items requested by customer $c \in C$. Each item $m \in M$ is defined by its length l_m, width w_m and a service time st_m which is the length of time required to either load or unload the item. An unlimited fleet of homogeneous vehicles is assumed to be available for each echelon. Without loss of generality, let us consider a finite set of vehicles to be $K = K^1 \cup K^2$, where K^1 and K^2 are the sets of vehicles associated with the first and second echelons, respectively. The number of items which must be delivered in each echelon serve as upper bounds for the number of vehicles $|K^1|$ and $|K^2|$, which will always be sufficient to deliver all items. Each vehicle has a rectangular loading surface where (un)loading operations can only be performed via the vehicle's rear. Vehicles belonging to the first/second echelon have a fixed cost h^1/h^2 and two dimensions: length l_{v1}/l_{v2} and width w_{v1}/w_{v2}. The length and width of vehicles and items are provided in centimeters. Based on the information provided by the logistics company, we can safely assume that physically (un)loading and maneuvering items at facilities does not pose a problem. Therefore, only the area occupied by items is considered when calculating the capacity usage at facilities. Thus, the loading constraints are only applied to the loading of items into vehicles.

A solution for the 2E-LRP2L is given by a finite vector $(\mathbf{R}, \mathbf{L})$, where $R = \{R_1, ...R_{|K|}\}$ and $L = \{L_1, ...L_{|K|}\}$. For each vehicle $k \in K$, R_k is a route performed by k and L_k is the full load plan of k as it departs from the platform/satellite. A route is deemed feasible if it adheres to the following constraints defined by Boccia et al. (2010): first-echelon routes start from a platform, serve one or more satellites and end at the same platform; second-echelon routes start from a satellite, serve one or more customers and end at the same satellite; and a customer must be served by a single satellite and a single vehicle. In the particular real-world application which has inspired this work, satellites have a far greater capacity than first-echelon vehicles. Therefore, the total demand allocated to a satellite may require multiple deliveries from a platform. Given this real-world demand we propose a relaxation of the standard 2E-LRP where split deliveries are allowed in the first echelon, thus a satellite may be served by multiple platforms and vehicles.

In order to mathematically represent L_k, we can consider the loading surface of a vehicle as the first quadrant of the Cartesian coordinate system bounded from above on one axis by the width and on the other axis by the length of the vehicle. The items to be delivered during R_k are represented by rectangles which must be positioned on this bounded region and are subject to the following loading constraints: all items assigned to a vehicle must fit totally within the loading surface of that vehicle; stacking is strictly forbidden and items must not overlap with one another; orthogonal loading is required (items have a fixed orientation and may not be rotated when loaded into a vehicle). Thus, the length edge of items must run parallel to the length edge of vehicles.

These loading constraints are visualized by way of Figure 2, which provides a bird's-eye view of the vehicle loading surface. The orientation of the items to be loaded is shown in Figure 2(a). This figure demonstrates a violation given that one of the items breaches the limits of the vehicle's loading surface. Figure 2(b) demonstrates an overlapping violation, with two items sharing a portion of the loading surface. In Figure 2(c) the rightmost item was rotated in order to fit into the vehicle. However, rotation is not allowed and therefore this item violates the orthogonal loading constraint. Finally, Figure 2(d) illustrates a load plan where no loading constraints are violated.

In addition to the aforementioned loading constraints, the items' loading sequence plays a crucial role in the 2E-LRP2L. The literature defines *sequential loading* as follows: at each customer i, items M_i must be unloaded in a single movement without the need to rearrange any of the items assigned to customers later in the route. In other words, the space between the rear door of the vehicle and all requested items M_i should be free of any item in $M \setminus M_i$. In the 2E-LRP2L, sequential loading is always preferable. Violations of this constraint mean that a certain item $m \in M \setminus M_i$ is

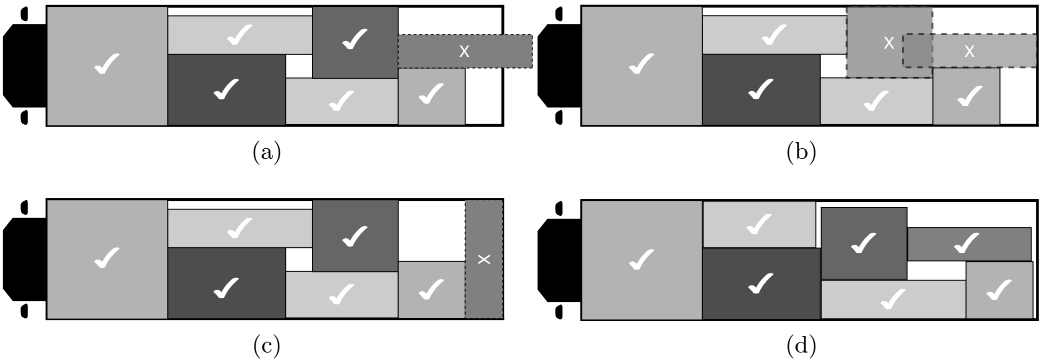


Figure 2: Loading constraints and violations.

blocking the unloading of another item $m' \in M_i$, which incurs a penalty proportional to the time it takes to rearrange m. Since one must not only unload but also reload an item which is blocking another, the penalty for moving item m is twice its service time ($2 \times st_m$). This penalty is referred to as a *sequential loading penalty* (Iori and Martello, 2010), or SLP for short, and is only applied to routes in the second echelon since we may face parking space and parking time restrictions at customer locations. Furthermore, most customers do not have the necessary equipment to (un)load every sort of item. By contrast, the sole purpose of satellites is to provide the necessary (un)loading facilities.

The objective of the 2E-LRP2L is to find a finite vector $(\mathbf{R}, \mathbf{L})$ serving all the customer demand while satisfying all the aforementioned constraints such that the total cost of (*i*) open facilities, (*ii*) routes, (*iii*) employed vehicles and (*iv*) sequential loading penalties is minimized. An example of how to calculate the cost of routes and possible penalties is shown in Figure 3. Recall that the items should be unloaded from rear (the rightmost side on the figure). Items 1, 2, 3 and 4 have service times equal to 10, 20, 5 and 14 minutes, respectively. Route (a) employing the load plan shown in (c) (pair a-c) and route (b) employing the load plan in (d) (pair b-d) both execute all their deliveries without needing to rearrange items. Therefore, these pairs do not incur any SLP cost. The routing cost is 105 for route (a) and 87 for route (b). However, if we consider the other two potential pairs, a-d and b-c, the costs will increase. In pair a-d, item 2 requires delivery first, but it is blocked by item 1. Thus, on top of the routing cost 105, pair a-d incurs an SLP cost of 20 due to the need to move item 1. Similarly, pair b-c delivers item 1 first, but it is blocked by item 2. Pair b-c incurs an additional SLP cost of 40 for moving item 2 while its routing cost remains 87.

Modeling challenges: In order to develop a mathematical formulation for the 2E-LRP2L one could extend the 2E-LRP formulation proposed by Boccia et al. (2011). This one-index formulation, provided in Appendix A, is a path-based formulation which defines a variable for each feasible route. In the 2E-LRP2L, routes must deliver a set of unstackable items M_c to each customer $c \in C$. Thus, all feasible routes must be generated considering two-dimensional loading constraints and sequential

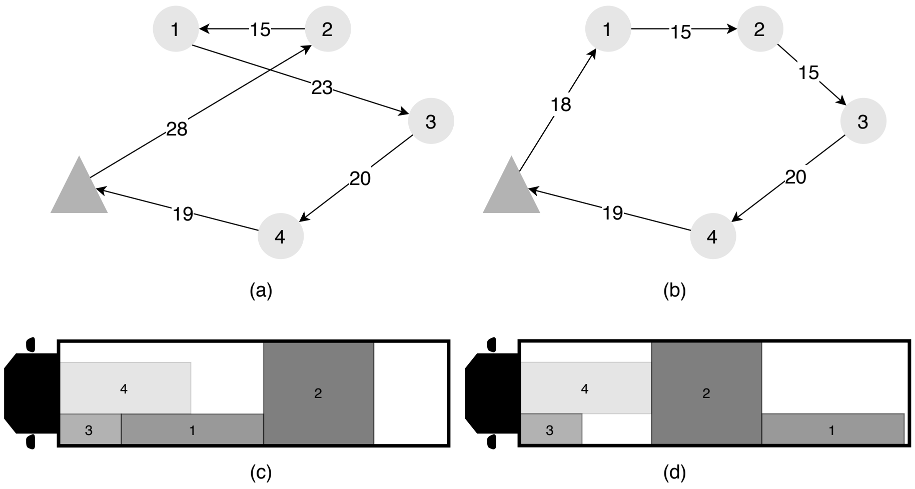


Figure 3: Cost for the combinations of two routes and two load plans.

loading penalties. Moreover, split deliveries must be included, but are only allowed for first-echelon routes. Given these nuances, there are two main adaptations one must make to the formulation by Boccia et al. (2011) in order to arrive at a formulation for the 2E-LRP2L.

First, when generating routes, the feasibility of each could be checked by employing the two-dimensional orthogonal packing with sequential loading constraint formulation by Côté et al. (2014). Since violations of sequential loading are penalized rather than strictly forbidden in the 2E-LRP2L, the constraints which enforce sequential loading in the formulation proposed by Côté et al. (2014) should be relaxed and included in the objective function with the corresponding penalties. Second, instead of customer requests (given by a set of items), first-echelon routes must deliver individual items $m \in M$ in order to facilitate split deliveries. Constraints which enforce a satellite to be served by only one vehicle should therefore be removed from the formulation by Boccia et al. (2011). However, since split delivery is not allowed in the second echelon, a single vehicle carrying all items $m \in M_c$ of customer c must depart from one satellite to serve the customer in a single trip. Therefore, satellites can be visited multiple times by vehicles departing from different platforms so long as all items $m \in M_c$ are delivered to the same satellite. To model such constraints one could introduce an additional variable ω_{mst} which is equal to 1 if item $m \in M$ is delivered to satellite $s \in S$ by first-echelon route $t \in \mathcal{T}^1$, and 0 otherwise.

This strategy for developing a mathematical formulation for the 2E-LRP2L represents an extension of existing methods. Although intuitive, the resulting formulation would be lengthy since, as we have shown, it requires one to combine multiple formulations and would involve a large number of variables and constraints. Indeed, given that the formulations by Boccia et al. (2011) already face challenges to solve even small-scale instances it is logical to assume that an extended version would experience even greater difficulties. Therefore, we will focus on developing a heuristic approach for solving the 2E-LRP2L.

4. Methodology

Inspired by the publications reviewed in Section 2.1, the method developed in this paper decomposes the problem into two LRPs, applies a bottom-up approach with a different methodology for each echelon and optimizes the location of satellites and routes with tailored heuristics. We implement an adaptation of Iterated Local Search (Lourenço et al., 2003) for the 2E-LRP2L, henceforth referred to as ILS-LR2L. At each iteration the incumbent solution undergoes a perturbation procedure and a locally optimal solution is obtained by applying a local search operator.

Since location changes are disruptive, the ILS-LR2L perturbation operator consists of a location phase which explores several satellite configurations. When location and routing heuristics are applied interchangeably (without a specific order), facility configurations may be generated without optimizing the routes. This is because multiple location heuristics may be applied one after another without a routing heuristic. In this paper, for each generated satellite configuration (perturbed solution), the local search operator (routing phase) is employed, which optimizes the assignment of customers and routes. A two-stage approach during the routing phase is adopted where first the candidate route is generated and then this route's feasibility is checked by employing a given loading strategy.

The basic framework of ILS-LR2L is detailed in Algorithm 1. It begins by applying the routing local search to the initial solution and saving the best solution (s^*) found so far (lines 3-4). Next, an iterative procedure generates a neighboring solution s' by perturbing the satellite locations of incumbent solution s, applying a routing heuristic in the second echelon and a dedicated heuristic for solving the first echelon (lines 6-8). This neighbor solution s' replaces the best solution s^* if the value of s' is better than that of s^*. Every η iteration the incumbent solution s restarts from the best solution or is replaced by neighbor solution s' (lines 11-14). This procedure limits how much an incumbent solution can differ from the best solution. After *maxNumIter* iterations, the best solution is returned (line 15). The following subsections detail the algorithm's components.

4.1. Initial solution

The initial solution method decomposes the problem into two LRPs and builds the solution in a bottom-up manner. Therefore, this method begins by generating a solution for the second-echelon LRP. First, candidate satellites are randomly opened until their cumulative capacity is capable of serving the total demand. Next, customers are selected in a random order and assigned to the closest open satellite with sufficient residual capacity, that is, the original capacity minus the total demand of the customers previously assigned to this satellite. We adapted the Clarke and Wright (1964) *savings algorithm*, initially proposed for VRPs, to produce feasible routes for the 2L-CVRP. Given a loading strategy, the adapted *savings algorithm* is employed to generate routes for each satellite. To avoid poor initial solutions, ι solutions are produced from which the best is selected.

Using the second-echelon solution as input, the first-echelon solution can then be generated. This order is necessary since the first echelon depends on the satellite configuration and the demand

Algorithm 1: ILS-LR2L framework.

```
Input: initial solution $s^0$                                  // $s^0$ produced by Algorithm 2
Input: loading strategy $ls$, $maxNumIter$
$s \leftarrow$ routing local search($s^0$, $ls$)
$s^* \leftarrow s$
for $i \leftarrow 0$ to $maxNumIter$ do
    $s_l \leftarrow$ perturb location ($s$)
    $s_r \leftarrow$ routing local search($s_l$, $ls$)
    $s' \leftarrow$ solve the first echelon($s_r$)
    if $f(s') < f(s^*)$ then
        $s^* \leftarrow s'$
    if $i$ mod $\eta = 0$ then
        $s \leftarrow s^*$
    else
        $s \leftarrow s'$
return $s^*$
```

associated with each satellite. In order to accelerate the optimization for the first-echelon LRP, we reduce the search space of platform locations to explore as follows. Let $\overline{S}$ be the set of satellites opened and *demand*(j) denote the total demand (area) allocated to satellite $j \in \overline{S}$. Let N_P^{min} denote the minimum number of platforms needed to serve $\overline{S}$. We then only consider the subsets $\overline{P} \subset P$ with cardinality $|\overline{P}| \in \{N_P^{min}, N_P^{min}+1\}$. Consider an instance with five potential platform locations. If a minimum of two platforms is necessary to meet the total demand, then all combinations of two and three platforms will be considered. The motivation behind considering subsets of P with cardinality $N_P^{min}+1$ is the likelihood of compensation for the cost of an additional platform with shorter routes. Preliminary experiments showed no improvements when adding more than one extra platform.

Given a platform configuration, the remainder of the initial solution method is similar to that used for the second echelon. The *demand*(j) of each satellite $j \in \overline{S}$ is assigned to the closest platform with sufficient residual capacity. The route is then generated using the *savings algorithm*. Finally, a 2-SWAP local search operator is applied to each first-echelon solution and the best is incorporated into the complete initial solution. Algorithm 2 provides an overview of how the initial solution is generated. In this algorithm, for any set of facilities U, *demand*(U) denotes the total customer demand in or associated with U (denoted *demand*(j) if j is the only element in U) and *cumulativeCap*(U) denotes the cumulative capacity of the facilities in U.

The initial solution method for the first echelon builds near-optimal solutions and is used every time a change is made to the second-echelon solution. In an attempt to save processing time, first-echelon solutions are cached together with the set of open satellites and the items required per satellite. Whenever a first-echelon solution requires the same satellite configuration, the solution in

Algorithm 2: Initial solution framework.

```
Input: empty solution $s*$, loading strategy $ls$
$C \leftarrow$ Customers in random order
$P_c \leftarrow$ Configurations with $N_P^{min}$ and $N_P^{min}+1$ platforms and sufficient cumulative capacity to serve $demand(C)$
while number of solutions generated $< \iota$ do
    Randomly open satellites until $cumulativeCap(\overline{S}) \geq demand(C)$
    foreach $c \in C$ do
        Assign $c$ to the closest open satellite with sufficient residual capacity
    foreach $j \in \overline{S}$ do
        savings algorithm($j$, $ls$)
    foreach $p_c \in P_c$ do
        Open platforms in $p_c$
        foreach Pair(item, satellite) do
            Assign item to the closest open platform with sufficient residual capacity
        foreach $p \in p_c$ do
            savings algorithm($p$, $ls$)
        Apply the 2-SWAP operator on the first-echelon routes
        Save the current best solution
    Update the best solution $s*$
return $s^*$
```

the cache is retrieved. Either the entire solution is reused if the satellite configuration and the items per satellite are the same, or only its platform configuration is reused to construct a new solution. In the latter case, the method builds a new first-echelon solution for only one platform configuration instead of all possible configurations. This cache therefore considerably decreases processing time without any loss in solution quality.

The time complexity of generating all combinations of n platforms in subsets of size u, where u is a constant, is given by $O(n^{min(u,n-u)})$. Although this operation may be costly, it is important to note that in the specific problem context of this paper, the first echelon contains a maximum of five candidate platforms, enabling a complete search of these locations in reasonable time. To compare how much processing time is used in each echelon, we perform a preliminary test to ascertain the total time spent building first-echelon solutions throughout all iterations of the ILS-LR2L. The experiment show that at most 20% of the total running time is dedicated to generating solutions for the first echelon. Finally, since in the 2E-LRP variants the dimension of the second echelon is much greater than the first, the proposed first-echelon algorithm remains relevant for larger instances as the bottleneck would continue to be the second echelon.

4.2. Location phase

The location phase perturbs the solution by changing satellite locations. This phase utilizes three neighborhoods: (i) close an open satellite, (ii) open a closed satellite and (iii) close an open satellite while opening one or more closed satellites. Each time the location phase is invoked, one of these three neighborhood is randomly selected. The new satellite configuration in a neighborhood is only maintained if its lower bound (see Section 4.3) is not worse than the current best solution. In other words, any satellite configuration which leads to a lower bound greater than the cost of the best known solution is discarded from the search space and never visited again in further iterations of any location neighborhood. As a result, the search will not spend unnecessary processing time optimizing the assignment and routing of customers in a solution whose satellite configuration can never improve upon the current best solution. Each neighborhood works as follows.

The first neighborhood opens a random satellite and selects a subset of customers to be assigned to this satellite. The customers are selected in one of two ways: (1) customers that are closer to the new satellite than to their current one or (2) the closest customer to the new satellite along with customers in the same route that are closer to the new satellite than to their current one. Once selected, the assignment of customers is performed in a random order and ends when the satellite is full or all customers have been assigned. The routes associated with the new satellite are then created by means of the savings algorithm. This neighborhood disallows closing satellites. Thus, a customer can only be removed from a satellite that is serving two or more customers.

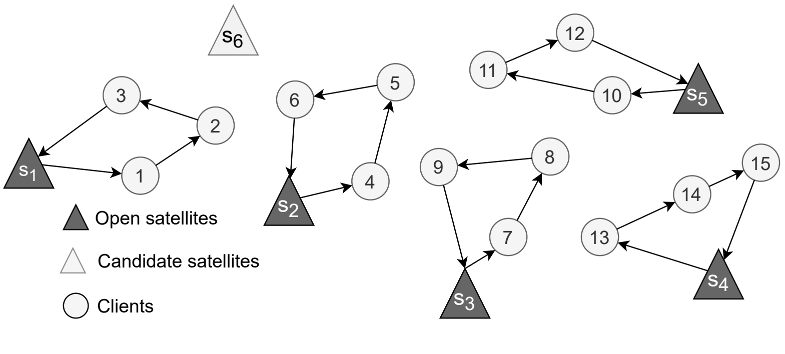
The second neighborhood closes a random satellite in such a way that the cumulative capacity of those remaining is still sufficient to meet the demand of all customers. This neighborhood tries to modify the existing routes as little as possible. First, the algorithm attempts to re-allocate entire routes to the best possible satellite while respecting capacity constraints. Whenever a complete route is unable to be reassigned, single customers are randomly inserted into the best position in their best possible route considering all satellites. An insertion position can be in a new or existing route.

The third neighborhood closes a random satellite and opens one or more satellites (depending on capacity requirements) that have the minimum average travel time from all removed customers. To avoid cycles, it is prohibited to reopen the most recently closed satellite. The insertion of customers is first conducted with respect to entire routes and then any remaining unassigned customers are inserted one by one into their best position.

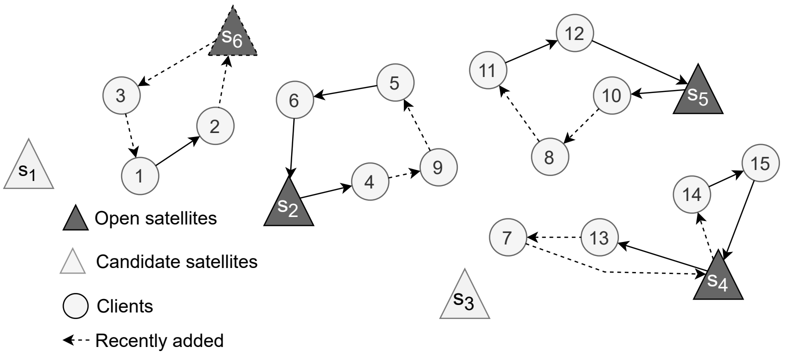
Figure 4 shows two second-echelon solutions to illustrate changes in satellite configuration. Figure 4(a) employs five satellites (s_1, s_2, s_3, s_4 and s_5) to serve all customers. Figure 4(b) presents an alternative solution after two moves have been applied. First, satellite s_3 has been closed and its customers (7, 8 and 9) have been reassigned to nearby satellites (s_2, s_4 and s_5). The second move swapped the status of satellites s_3 and s_6, with the complete route of s_1 reassigned to s_6.

4.3. Lower bound

The location of satellites is of major importance when it comes to obtaining good solutions. One can imagine that while certain configurations of open satellites might result in high-quality



(a)



(b)

Figure 4: Second-echelon routes with different satellite configurations.

solutions, others will consistently result in poor solutions. Thus, we design a mechanism that utilizes a lower bound function and is capable of identifying and avoiding poor satellite configurations.

Given a configuration of open satellites, the lower bound is calculated by summing together the cost of opening those satellites, the minimum vehicle cost and the minimum routing cost needed to serve satellites and customers. The minimum vehicle cost is given by the minimum number of vehicles needed in each echelon multiplied by their cost. The minimum number of vehicles is determined by the minimum length required to load all orders divided by the length of the vehicles.

Figure 5 illustrates how the minimum routing cost in the second echelon is calculated. A minimum spanning tree of all customers is first generated, see Figure 5(a). In Figure 5(b), this tree is then partitioned into clusters by removing the ℓ longest edges, where ℓ is the number of open satellites minus one. Each satellite is then connected to its two nearest customers (Figure 5(c)). The same procedure is conducted for the first echelon, where a single platform is opened and connected to the clusters of open satellites. The opened platform is the one which results in the lowest total cost considering platform opening cost and routing cost from the platform to satellites.

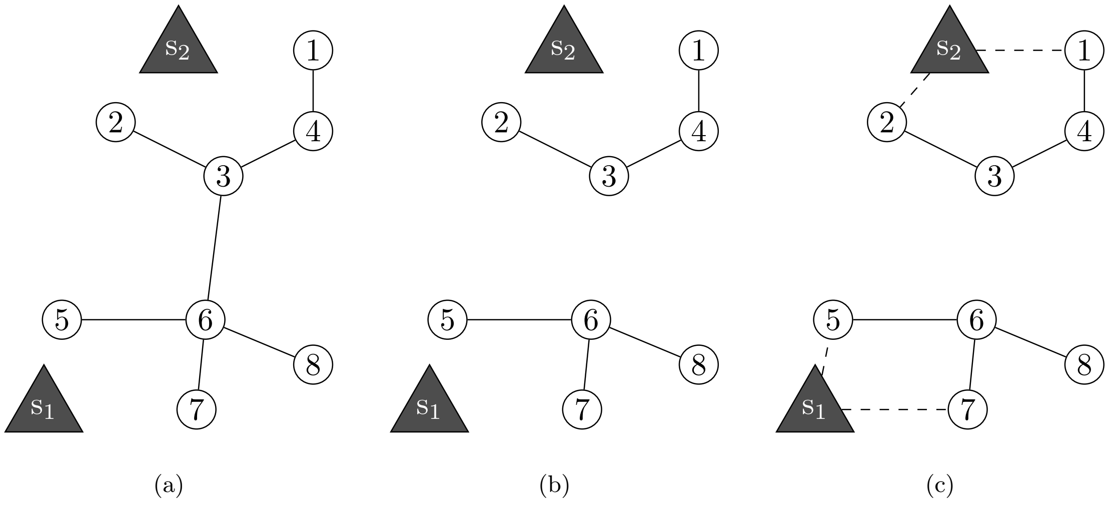


Figure 5: Second echelon minimum routing cost construction.

4.4. Routing phase

Given that the routing phase is only one of many components comprising ILS-LR2L, it must quickly produce high-quality solutions. One of the state-of-the-art heuristics for VRPs, referred to as SISRs, was introduced by Christiaens and Vanden Berghe (2020) and is adopted in this work. The ruin and recreate heuristic was chosen given its speed, simplicity to implement and proven efficiency in terms of the quality of generated solutions for a range of VRPs.

Christiaens and Vanden Berghe (2020) introduced a novel property, referred to as spatial slack, which removes a sufficient number of customers from different geographically proximate routes, creating spatial and capacity slack in each route. Introducing such slack provides a range of options when reassigning a customer, since multiple routes close to that customer might be available for insertion. Regarding the recreate operator, insertion positions are selected using rank-based probabilities, which prevents the best insertion method from converging in an excessively greedy manner.

This paper proposes a multi-depot adaptation of SISRs, henceforth referred to as SISRs-MD, which removes customers from different, yet geographically proximate, routes. Note that these routes may be served by different satellites. When reinserting customers, the best insertion procedure considers all satellites and routes. Each time SISRs-MD is called it performs a number of iterations and newly generated solutions are evaluated based on a simulated annealing acceptance criterion. The initial temperature value for the simulated annealing algorithm is solution-dependent and computed as $-wp \times S'/\log 0.5$ (Ropke and Pisinger, 2006), where S' is the value of the solution passed to the routing phase and wp is the worsening percentage which controls how much an accepted solution may be worse than the incumbent solution. This initial temperature states that a solution ($wp \times 100$)% worse than S' is accepted with 50% probability. With the exception of wp, SISRs-MD employs all the original parameter values by Christiaens and Vanden Berghe (2020) and returns an improved or equal quality solution.

Since the ruin method works based on geographic distances, it may happen that all customers removed in one iteration belong to the same satellite. Although improvements may still occur, the search would probably be too localized. Therefore, it is worthwhile developing additional operators to enforce the assignment of customers across different satellites.

The ruin and recreate methods for the SISRs-MD were extended to force, with a certain probability, the removal/insertion of customers from/into different satellites. In other words, during σ_{ruin}% of all SISRs-MD iterations a small set of customers are removed from every open satellite. In the same way, there is a $\sigma_{recreate}$% chance customers are inserted into their best position considering all satellites and a (100-$\sigma_{recreate}$)% chance they are inserted into their best position excluding the satellite from which they were just removed.

To further improve routes, at the end of the routing phase SISRs is applied to each satellite followed by a descent heuristic. We consider two descent heuristics: MOVE and 2-OPT*. The MOVE operator relocates a sequence of 1-3 customers to its best position in any route. The operator works intra- and inter-route as well as intra- and inter-satellite. Meanwhile, the 2-OPT* operator (Potvin and Rousseau, 1995) is only performed intra-satellite for every pair of routes u and v and every customer $i \in u$ and $j \in v$. Both the MOVE and 2-OPT* operators are performed in a random order and in a first-improvement manner. The two operators are applied repeatedly and the search ends when both operators return a solution with no improvement. These additional operators are commonly used for VRPs, cheap in computational cost and bring an improvement of 0.03% on average for the considered instance sets.

4.5. Load plan method

In order to create feasible load plans we apply an algorithm to determine the placement (position) of items inside vehicles. This load plan method (LPM) uses a placement heuristic and attempts to build a feasible load plan for a given vehicle size and set of items. We apply the best-fit placement heuristic proposed by Burke et al. (2004) and follow the efficient implementation by Imahori and Yagiura (2010). We choose this fast approach since it produces high-quality solutions and is simple to implement. The LPM can be called when a new customer or item is inserted into a route (by any of the local search operators) or when the visited order of customers is altered.

In order to produce a higher number of feasible load plans, the method by Burke et al. (2004) is extended to include additional strategies and insertion policies. The implementation in this paper sorts items based on their width, length and area. Ties between items with the same dimension are broken by either length or width. Items are sorted in reverse order of delivery with a view to achieving a low SLP value. Thus, the last item to be delivered is placed inside the vehicle first. When placing an item inside the vehicle, one of two insertion policies can be followed: insert the item next to the longest neighbor or insert the item next to the neighbor with the most similar length. The LPM combines each item order with an insertion policy and returns the first feasible load plan found.

Every time a load plan is generated, feasible or not, it is stored together with the route in a memory structure entitled Trie (Leung et al., 2010). Trie avoids evaluating routes that have already had their load plan checked. Thus, given a route, the method first searches for it in the Trie structure and, if it is not present there, only then is the LPM called and the structure updated with the new route and corresponding load plan.

4.6. Loading strategies

Different loading strategies are proposed to study the best way to handle the loading constraints associated with the 2E-LRP2L. By employing different strategies, this paper aims to investigate the impact a dedicated load plan method has on 2E-LRP2L solutions and to study how item loading sequences and SLP influences solution quality.

This first strategy is referred to as *LazyLoading* and attempts to mimic what is currently done in practice by the company that inspired this study. *LazyLoading* assumes that the loading of large items is simple and that a dedicated load plan method is unnecessary. Therefore, when constructing vehicle routes throughout ILS-LR2L iterations, only one-dimensional capacity constraints are enforced. In this case, items are loaded into vehicles as long as their accumulated area is less than the vehicle's total area. The LPM described in Section 4.5 is then applied only a single time in the final solution to generate a load plan and calculate the SLP.

During the preliminary tests, we observed that the ILS-LR2L with *LazyLoading* without any further adjustments resulted in infeasible load plans for 50% of the routes. More specifically, although the vehicle capacities were respected area-wise, in reality the items could not fit together inside the vehicle. Consider Figure 6, where the total area of the vehicle is 17.5 and the three items which require loading have a combined total area of 14.18. If one only considers area one would expect a feasible load plan, yet in reality a feasible load plan is impossible due to the dimensions of the items.

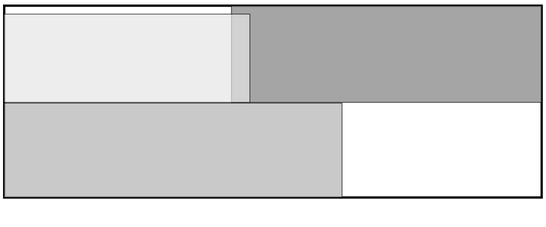


Figure 6: An infeasible load plan for items whose total area is less than that of the vehicle.

If a vehicle turns out to have an infeasible load plan, items will need to be unloaded and reloaded into other vehicles or additional vehicles may need to be employed. This may lead to additional costs and delays. As a quick remedy for this issue, route planners in the company artificially modify the vehicle capacities. More specifically: they employ reduced capacities in their *LazyLoading*-like strategy.

Such a quick fix increases the chance of generating feasible solutions without needing to validate the challenging two-dimensional loading constraints. Therefore, we impose a *maximum occupancy rate* on the *LazyLoading* strategy. $MaxL_{v1}$ and $MaxL_{v2}$ denote these rates for the first- and second-echelon vehicles, respectively. Although these restrictions significantly reduce the number of infeasible solutions, 3% of the routes *LazyLoading* generates using these maximum occupancy rates continue to be infeasible.

To guarantee route feasibility and evaluate the impact of the loading constraints on the final solution a different approach is taken. *ActiveLoading* employs the LPM for every generated route. This approach is necessary since, as demonstrated, capacity constraints do not guarantee the generation of feasible routes for the 2E-LRP2L. Additionally, to investigate the impact of the loading sequence of items in the solution, the *ActiveLoading* approach is divided into three loading strategies, where each is defined in accordance with how exactly they handle the SLP. The three strategies, ordered from least to most restrictive, can be summarized as follows:

- *LateSLP*: route feasibility is checked by the LPM but no SLP is applied until the final solution.
- *OnTimeSLP*: the SLP is calculated and added to the route cost on the fly at each load plan feasibility check.
- *StrictSLP*: the SLP is calculated at each load plan feasibility check and routes with non-zero SLP are considered infeasible. Note that this strategy is capable of providing guaranteed feasible solutions for an eventual 2E-LRP2L variant where sequential loading is mandatory.

These three *ActiveLoading* strategies result in lengthy computational times given that the LPM (calculating or not the SLP) is invoked for every generated route. However, it is unlikely that infeasible load plans are generated at the beginning of route construction since vehicles are largely empty and there is plenty of space to place items. Therefore, to reduce the processing time of the *ActiveLoading* strategies, we propose a *minimum occupancy rate* so that the LPM is only called if the occupancy rate is above $MinL_{v1}$ and $MinL_{v2}$ for first- and second-echelon vehicles, respectively.

5. Computational study

All experiments were conducted on a computer with an Intel Xeon E5-2660 processor at 2.6 GHz, with 164 GB of RAM running Ubuntu 18.04 LTS. ILS-LR2L was implemented in C++ and compiled using gcc 7.4.0 and option -O3. All instances, solutions and tables with results are available in a public repository, i.e., Mendeley Data (`link.suppressed.due.to.double.blinded.review`).

Table 2 details all necessary parameters for implementing ILS-LR2L and their respective values. Parameters concerning the number of iterations were calibrated manually with a compromise between solution quality and processing time. The other parameters were calibrated using *irace* (López-Ibáñez et al., 2016), with the range of values provided to irace also shown in Table 2.

Table 2: ILS-LR2L parameters and values.

Parameter	Value	Parameter	Value	Range
$maxNumIter$	50	wp	0.003	{0.001, 0.003, 0.005, 0.01, 0.03}
SISRs-MD iterations	1000	η	2	{1, 2, 4, 6, 8, 10}
SISRs iterations	500	$\sigma_{recreate}$	60	{20, ...,80}
		σ_{ruin}	50	{20, ...,80}
		ι	10	{1, 5, 10, 25, 50, 75, 100}
		$MaxL_{v1}$	80%	{50, ..., 100}
		$MaxL_{v2}$	70%	{50, ..., 100}
		$MinL_{v1}$	70%	{50, ..., 95}
		$MinL_{v2}$	60%	{50, ..., 95}

During preliminary experiments, the maximum occupancy rate parameters were calibrated in such way that no infeasible route is generated when the occupancy rate is under $MinL_{v1}$ and $MinL_{v2}$. When employing these minimum occupancy rates, the processing times for the *ActiveLoading* strategies are eight times faster, with the average processing time plummeting from 3768 seconds to 470 seconds without any loss in solution quality. It is important to note that for all the experiments in this section, $MaxL_{v1}$ and $MaxL_{v2}$ are always enforced when using the *LazyLoading* strategy, while $MinL_{v1}$ and $MinL_{v2}$ are always enforced when using one of the three *ActiveLoading* strategies.

5.1. New instance set

To run experiments with ILS-LR2L, and in order to stimulate further research regarding the 2E-LRP2L, instances were generated based on real-world data. A transportation company of large equipment provided locations of facilities (platforms and satellites) and customers, vehicle and items sizes (l_v, w_v and l_m, w_m), and approximately two months worth of customer demands and their respective day of delivery. There are up to four platforms and eight satellites which can be operational depending on the total customer demand. Vehicles have equal width (2.5 meters) but three different lengths (17.6, 8.5 and 7 meters), while items are at most 5 meters long and 2.5 meters wide.

A travel time matrix provides realistic travel times between all pairs of locations. Meanwhile, service time st_m is based on item m's dimensions and may range from 5 to 20 minutes. Requests were divided into 32 instances according to their delivery date and range from 81 to 274 customer requests. Due to privacy issues, all identifying features such as names and physical locations have been omitted.

The cost and capacity of facilities and the cost associated with vehicles were not provided by the company and were generated for this paper in accordance with the 2E-LRP benchmark instances.

We first analyzed how the cost is distributed among the many components present in a 2E-LRP solution. Next, costs were calibrated to generate 2E-LRP2L instances with similar proportions. The cost breakdown is illustrated in Appendix C (Figure C.11), where the total cost of an initial solution is divided in percentages considering first and second echelons, the cost of open facilities and routes, as well as traveling time and vehicle cost. The reported percentages are an average across all instances.

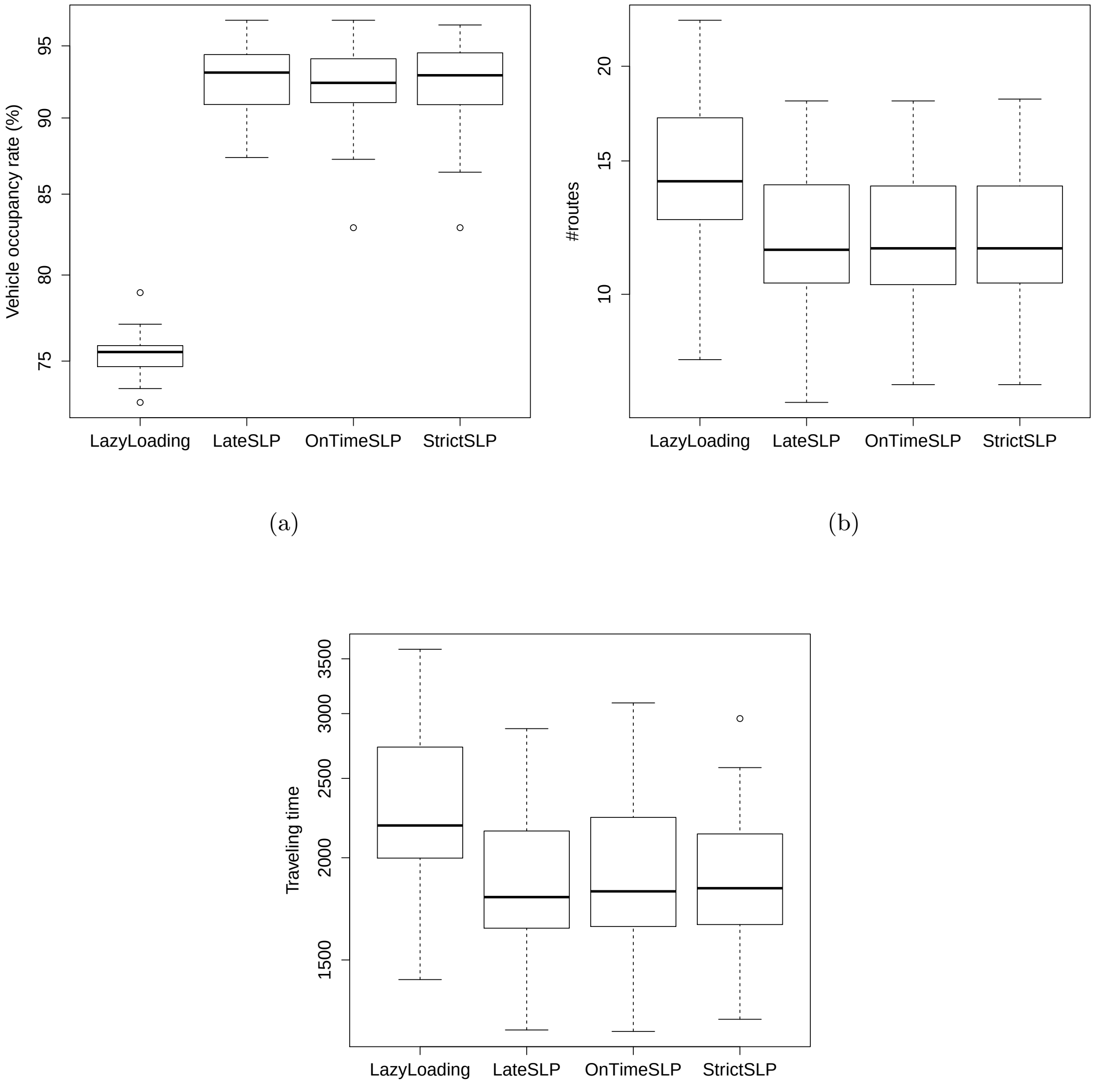
Tuning facility capacities correctly is important in order to create instances that represent the range of different sizes observed in practice. The idea is to assign capacities to facilities in such a way that all customers may be served without needing to open all facilities. For each instance, given a set of customers and their respective demand, the total area of all requested items was taken into account when generating the capacity of facilities. Satellite capacities were randomly generated between 17% and 30% of total item area. In this way, solutions must open at least four satellites out of the eight available. Platform capacities were selected between 40% and 70% of total item area, which results in solutions with two or three open platforms out of four.

5.2. Detailed results

This section presents computational experiments and compares the results produced by ILS-LR2L using the four different loading strategies. Figures 7(a)-10(a) provide boxplot charts which compare the four loading strategies, namely: *LazyLoading*, *LateSLP*, *OnTimeSLP* and *StrictSLP*. The values in the charts correspond to the average values of 10 runs across all 32 instances. To evaluate statistically significant differences, the pairwise T-test was performed with a confidence level of 95%.

Figures 7 provides computational results concerning the first echelon. The *LazyLoading* strategy, which employs a maximum occupancy rate of 80% for the first-echelon vehicles, uses less vehicle capacity than the other strategies. As a result, one can observe that more routes and longer traveling times are needed. More specifically: *LazyLoading* employs 22.8% more routes on average and travels 20.3% longer. The pairwise T-test shows that while all three *ActiveLoading* strategies perform similarly, given that no SLP is applied in the first echelon, statistically significant differences are observed between each *ActiveLoading* strategy and *LazyLoading*.

Figure 8 provides the results of the second echelon, where the SLP is applied. For the *ActiveLoading* strategies, Figure 8(a) shows a decrease concerning vehicle occupancy rates as the loading constraints become more restrictive. However, *LazyLoading* is a clear outlier here due to the aforementioned maximum occupancy rate restrictions (which are set to 70% for second-echelon vehicles). A similar discrepancy can also be observed in Figure 8(b), where *LazyLoading* uses 34.92% more routes than *StrictSLP*. Statistically significant differences are observed for every pair of strategies with the exception of pair *OnTimeSLP*-*StrictSLP*. In other words, for the *ActiveLoading* strategies, *LateSLP* uses significantly fewer routes with more load per vehicle than *OnTimeSLP* and *StrictSLP*.



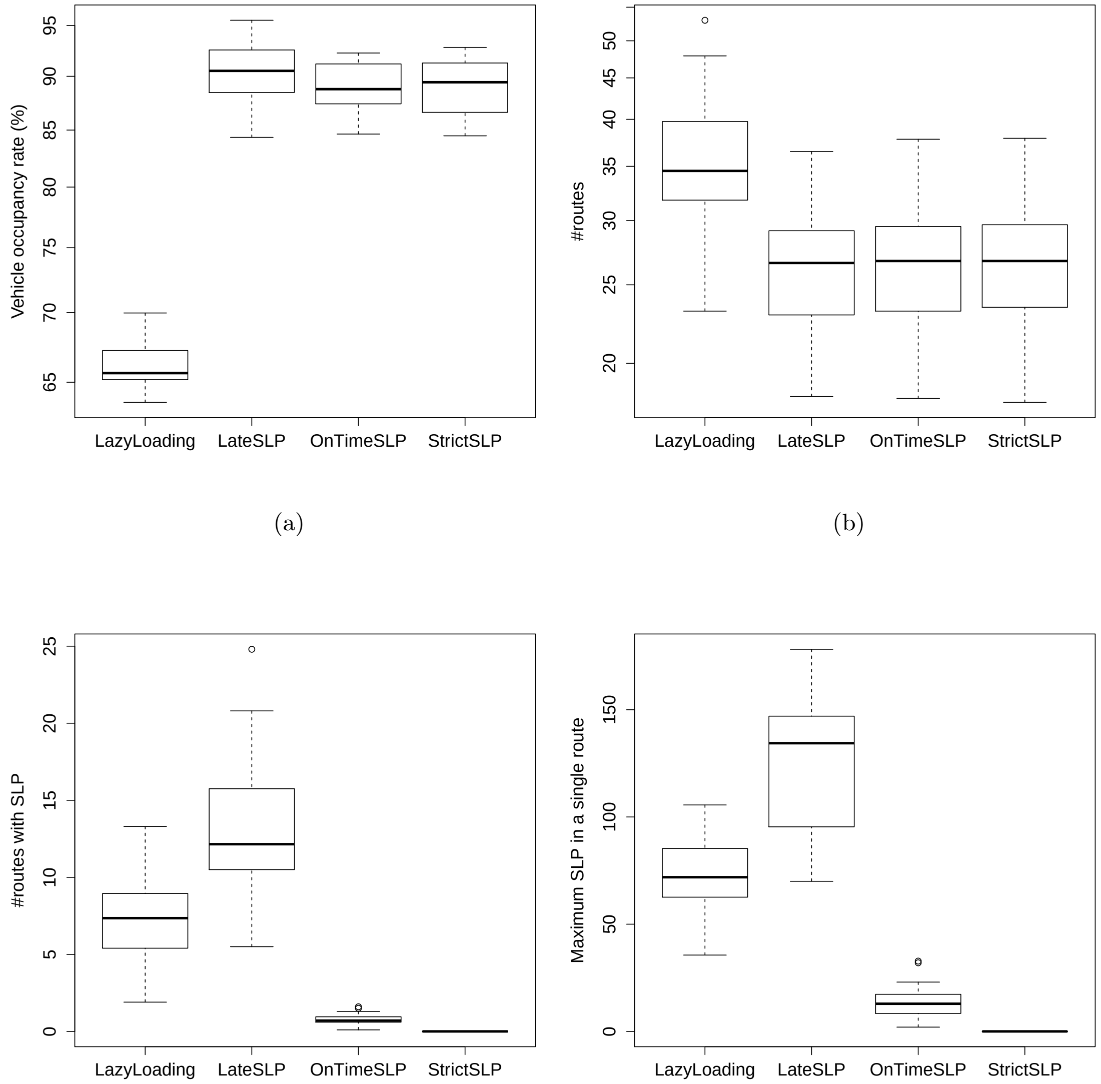
(a)

(b)

(c)

Figure 7: Different loading strategies for the first echelon.

Figure 8(c) shows the proportion of routes which incur penalties per loading strategy. Only 2.9% of the routes produced by *OnTimeSLP* incur SLP penalties, while *StrictSLP* forbids them totally. A significant increase can be observed when the SLP is not applied during the routing phase, as 19% and 48% of the routes produced by *LazyLoading* and *LateSLP*, respectively, incur a penalty. More penalties are observed in *LateSLP*'s routes than in *LazyLoading*'s due to the fact that the vehicles of *LateSLP* have much higher occupancy rates. Thus, the more items a load plan has,



(a)

(b)

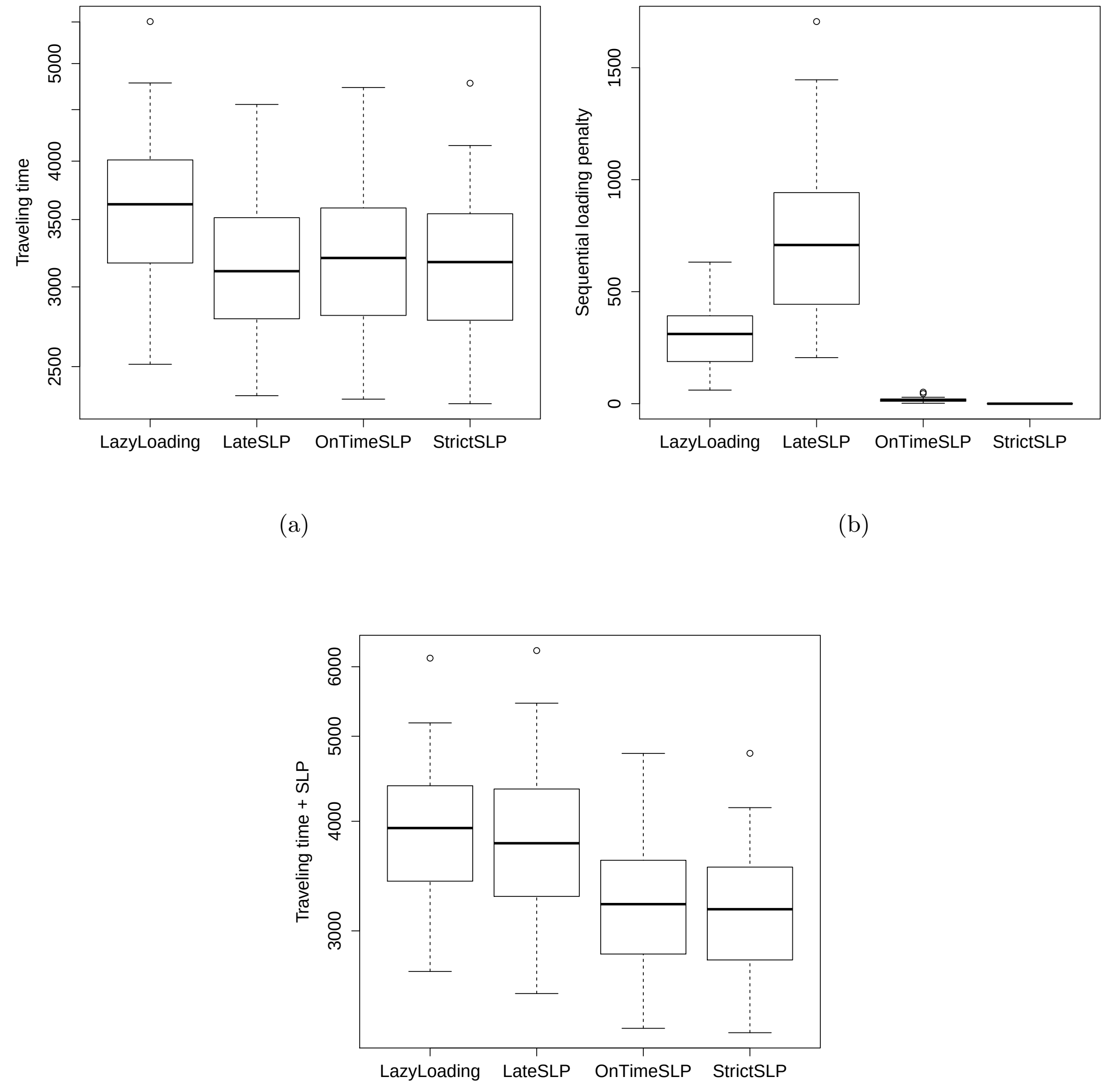
(c)

(d)

Figure 8: Different loading strategies for the second echelon.

the greater the chance sequential loading is violated when it is not explicitly checked for. The total penalty associated with a single route may reach as high as 178 minutes (see Figure 8(d)), which is a very long idle time spent at the customer's location when one considers that the average driving time for a route is only 115 minutes. Such results imply that poor loading sequences can result in vehicles spending upwards of half their service time parked at customer locations rearranging items.

Figure 9 provides the total traveling time and the total penalty incurred by the second echelon. The *LazyLoading* and *LateSLP* strategies take up to 20% longer to complete routes. This additional time is strongly related to the penalties their routes incur. When comparing only the traveling time, statistical tests demonstrate significant differences between every pair of strategies, with the exception of *OnTimeSLP*-*StrictSLP*. With respect to the total time spent, the traveling time with SLP (Figure 9(c)) results show that all pairs are significantly different, with the exception of *LazyLoading*-*LateSLP*.



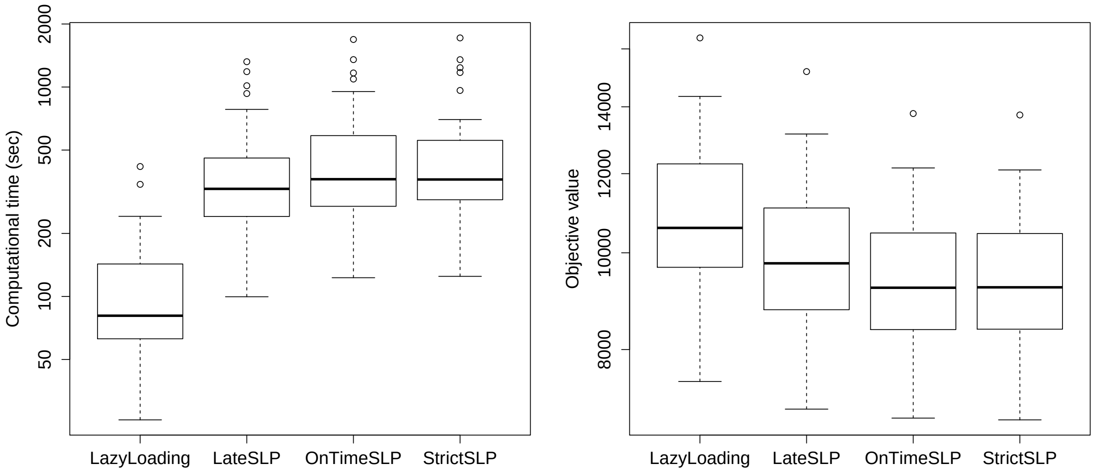
(a)

(b)

(c)

Figure 9: Different loading strategies for the second echelon. Traveling time and penalty analysis.

Finally, Figure 10 illustrates the average processing time and the best solution found across all 10 runs. Since *LazyLoading* does not employ the LPM, it is considerably faster than the *ActiveLoading* strategies. As for solution quality, *StrictSLP* performs best, generating 21 of the best-known solutions with only a 0.12% gap with respect to the best solutions found by the other strategies. *OnTimeSLP* obtained the remaining 11 best-known solutions and exhibits a gap of 0.29%. *LateSLP* and *LazyLoading* clearly have the worst average solutions and exhibit gaps of 5.45% and 15.42%, respectively. For both processing time and best solution values, there are statistically significant differences between all pairs of strategies, with the exception of *OnTimeSLP*-*StrictSLP*.



(a) (b)

Figure 10: Best solution comparison and run time of different loading strategies.

5.3. Result discussion

Even though *LazyLoading* is considerably faster than the three *ActiveLoading* strategies, it is associated with more routes and substantial penalties which translate into far higher total costs. The poor performance of the *LazyLoading* strategy in terms of solution quality indicates the necessity of applying a dedicated load plan method during the optimization process.

When dealing with large items, the time spent moving (loading and unloading) such items may be significant when compared to the average travel time. Considering the three *ActiveLoading* strategies, *LateSLP* results in slightly shorter travel times than *OnTimeSLP* and *StrictSLP*. However, *LateSLP*'s poor final solution quality demonstrates how SLPs may negatively influence solution values when they are ignored. *OnTimeSLP* and *StrictSLP* consider the sequential loading penalty at all times and, as a result, produced the best solutions. *OnTimeSLP* efficiently keeps penalties at a low level and generates good quality solutions. The similarity between *OnTimeSLP* and *StrictSLP* is noticeable, highlighting the impact of the SLP on the final solution in the studied context: that of large items whose rearrangements are time-consuming compared to travel times.

StrictSLP not only produces high-quality solutions for the present problem, but it can also be used for problem environments in which sequential loading is mandatory. This situation occurs when customers do not possess the unloading equipment necessary to move other customers' items, when physical space at the customer locations is limited or when parking time limits are in effect.

6. The 2E-LRP special case

This section is dedicated to strengthen the computational experiments by using benchmark instances available in the literature. Given the absence of instances for the 2E-LRP2L, the quality of our heuristic is evaluated by solving the 2E-LRP special case. In order to eliminate unnecessary loading constraints and split-delivery procedures and improve the efficiency of the algorithm when solving this special case problem, we initially present a few modifications of ILS-LR2L. The results obtained from this modified algorithm, which we refer to as ILS-LR, are then compared against those generated by state-of-the-art methods and statistical tests are conducted to evaluate and verify the competitiveness and quality of ILS-LR.

In the 2E-LRP, the capacity of platforms and satellites is given by Q_i ($i \in P \cup S$). A demand $D_c > 0$ is associated with each customer $c \in C$. Vehicles in the first and second echelons have capacity q^1 and q^2 and a fixed cost h^1 and h^2 is incurred when such vehicles are routed. The feasibility of facilities and routes relies only on the total capacity being respected. The 2E-LRP does not allow split deliveries to take place in either echelon and therefore customers and open satellites must both be served by a single vehicle.

6.1. Benchmark instances

This computational study employs three benchmark sets, which are referred to as *Prodhon*, *Nguyen*, and *Sterle* and contain a total of 147 instances. The first two instance sets were introduced by Nguyen et al. (2012b) and contain only one platform in the first echelon. *Prodhon* comprises of 30 instances arising from the CLRP with the addition of a single platform at coordinates (0,0). The instances range in size from 20-200 customers and 5-10 satellites. The second set, *Nguyen*, comprises of 24 instances containing 25-200 customers and 5-10 satellites. The third and final instance set, *Sterle*, was generated following the specifications outlined by Boccia et al. (2010) and made available by Contardo et al. (2012). The instance set is composed of 93 instances divided into three groups (I_1, I_2 and I_3) with different spatial distributions of satellites. Instances contain 8-200 customers, 3-20 satellites and 2-5 platforms.

6.2. ILS-LR for the 2E-LRP

The initial solution for both echelons employs the heuristic described in Section 4.1, with the exception that split delivery is not allowed. As a result, a platform must serve all items requested by a given satellite. The location and routing phases are also conducted as before. Meanwhile, the lower bound is calculated slightly differently: the minimum number of required vehicles is given by the total demand of goods divided by vehicle capacity.

The final solution of the 2E-LRP undergoes a post-processing step consisting of an exact approach for the first echelon. Since first-echelon dimensions are generally small, with few platforms and routes the formulation can be solved within short computational runtimes. For instances with only one platform, an optimal solution is generated using the Capacitated Vehicle Routing Problem formulation introduced by Kulkarni and Bhave (1985). For first-echelon problems with multiple platforms, we propose the Capacitated Multi-Depot Location Routing Problem (CMLRP) formulation in Appendix B. This CMLRP model uses load conservation constraints for sub-tour elimination (routes disjoint from P) and determines both the platform locations and the routes from those platforms to the given satellites.

For the 2E-LRP benchmark considered in this work, the mathematical formulations are on average responsible for a marginal increase in time of 23 seconds per instance set while providing an improvement of 0.02% in the solution value. This post-processing step guarantees optimal CVRP/CMLRP solutions for the first echelon with short processing time and can be valuable for future sets of instances of a similar size. Since these mathematical formulations are a new component in the methodology for the 2E-LRP, which was not used in the 2E-LRP2L method, we performed statistical tests to analyze their influence on solution value. These tests revealed no significant difference between ILS-LR approaches with and without the mathematical formulations on the current sets of instances. Therefore, we remain confident of the quality and competitiveness of ILS-LR2L, even without an exact approach as post-processing for the first echelon.

6.3. Parameters and setup

Experiments were performed using the same computational architecture and setup detailed in Section 5. The values of all the calibrated parameters were maintained. The only difference lies in the algorithmic adaptations detailed in Section 6.2 and the number of iterations, which was increased given that the ILS-LR is much faster without the loading constraints (see Table 3).

Table 3: Parameters and respective values for the ILS-LR.

Parameter	Value
$maxNumIter$	150
SISRs-MD iterations	15000
SISRs iterations	5000

6.4. Comparison with the state-of-the-art 2E-LRP methods

Table 4 compares ILS-LR against two state-of-the-art algorithms: ALNS (Contardo et al., 2012) and VNS (Schwengerer et al., 2012). Due to the algorithms' stochastic component, 10 runs were performed for each instance. Gap_{min} from Table 4 refers to the percentage difference between the best value s obtained from all runs and the best value published $s*$. $Gap_{s,s*}$ is calculated as $100 \times \frac{s-s*}{s*}$. Meanwhile, Gap_{avg} corresponds to the percentage difference between the average best solution, considering all the runs, and the best-known solution. The average CPU time in seconds is provided in column t_{avg}.

According to Passmark (2020), the hardware we employed is 1.37 and 1.50 times faster than the hardware employed by Contardo et al. (2012) and Schwengerer et al. (2012), respectively. Therefore, runtimes have been converted so as to enable a fair comparison (original runtime from ALNS and VNS divided by 1.37 and 1.50, respectively). The number of best-known solutions found by each algorithm is detailed in the last row of the table.

Table 4: Comparison of state-of-the-art methods. Best results of ILS-LR are shown in bold.

	ALNS			VNS			ILS-LR		
	Gap_{min}	Gap_{avg}	t_{avg}	Gap_{min}	Gap_{avg}	t_{avg}	Gap_{min}	Gap_{avg}	t_{avg}
Prodhon	0.32	0.83	339.17	0.08	0.50	208.69	0.08	0.27	134.30
Nguyen	0.16	0.49	139.78	0.27	0.90	183.08	**-0.03**	0.29	161.12
Sterle I1	0.24	0.41	223.36	0.05	0.60	222.98	**-0.02**	0.20	134.05
Sterle I2	0.25	0.45	241.01	0.25	0.72	202.65	**0.11**	0.32	197.40
Sterle I3	0.05	0.22	240.17	0.04	0.35	191.70	**0.02**	0.25	192.67
# of BKS(147)	115			115			127		

ILS-LR obtains 75.51% of the best-known solutions and improves upon an additional 16 instances (10.88%). Regarding the average gap, ILS-LR outperforms the previous algorithms across almost all instance sets in terms of both best solutions and average solution values. The average gap produced by the ILS-LR is the lowest of all methods, except for the best solution for the Prodhon instance set and the average solution for the Sterle I3 instance set.

A pairwise T-test was performed to assess the results of the three algorithms for each instance set. Table 5 reports these results, with statistically significant differences highlighted in bold. With a confidence level of 95%, ILS-LR dominates previous state-of-the-art methods for the Sterle instance set and is clearly competitive for the other two.

Table 5: Pairwise T-test results.

	Prodhon		Nguyen		Sterle	
	ALNS	ILS-LR	ALNS	ILS-LR	ALNS	ILS-LR
ILS-LR	**0.011**	-	0.12	-	**0.036**	-
VNS	**0.031**	1.00	1.00	0.08	0.426	**0.042**

7. Conclusions

The two-echelon location-routing problem (2E-LRP) considers freight distribution at two levels, where decisions include choosing the location of facilities and routing from facilities to customers.

This challenging problem has been widely studied and a number of methods are available. However, in practice, transportation companies are faced with additional complex decisions regarding the loading of items.

In order to supply such companies with feasible solutions, this paper introduced the two-echelon location-routing problem with two-dimensional loading constraints (2E-LRP2L) and a heuristic optimization method named ILS-LR2L. With this heuristic, several loading strategies, concerning how to handle the loading of items, were investigated using a set of instances based on real-world data. These instances have been made publicly available online to stimulate future research on the 2E-LRP2L.

This paper revealed that a loading strategy which considers only one-dimensional capacity constraints during optimization may lead to many infeasible routes, even when considering small numbers of large items. Results indicate that a dedicated load plan method, which considers two-dimensional loading constraints, considerably increases the occupancy rate of vehicles and thus reduces the number of necessary routes. Moreover, if moving items is time-consuming, optimizing the loading sequence of such items helps reduce route duration by as much as 20%.

With minor modifications, ILS-LR2L was also able to address the 2E-LRP. Computational experiments demonstrated that this modified version of the algorithm is competitive, producing a higher number of best-known solutions than previous state-of-the-art methods, while also generating some new best solutions. Statistical tests show how the ILS-LR2L dominates previous approaches for at least one instance set.

In terms of future research, this study provides a basic methodology for investigating the impact of real-world constraints on vehicle routing solutions. For approaches considering two-dimensional loading constraints, it is strongly recommended to define the minimum vehicle occupancy rate presented in this paper, as it is able to drastically reduce the processing time. The loading constraints could further be generalized to handle three-dimensional stacking.

Acknowledgements

This research was supported by Internal Funds KU Leuven and by the Research Foundation Flanders (FWO) Strategic Basic Research project Data-driven logistics (S007318N). It was also supported by the National Counsel of Technological and Scientific Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq) 408341/2018-1, Brazil. Editorial consultation provided by Luke Connolly (KU Leuven).

Appendix A.

We present the one-index formulation for the 2E-LRP proposed by Boccia et al. (2011) for a clear understanding of the constraints. This path-based formulation could easily be adapted to a simplified version of the 2E-LRP2L and comprises of the following parameters and variables:

Parameters:

D_c is the demand of customer $c \in C$.

Q_i is the capacity of depot $i \in P \cup S$.

q^1 and q^2 are the capacities of first- and second-echelon vehicles, respectively.

$\mathcal{T}^1$ and $\mathcal{T}^2$ are the sets of routes for the first and second echelons, respectively.

$\mathcal{T}^1_p$ is the set of routes starting from platform p ($\mathcal{T}^1_p \subseteq \mathcal{T}^1$).

π_i is the total cost of route $i \in \mathcal{T}^1 \cup \mathcal{T}^2$.

$\alpha_{is} = 1$ if satellite $s \in S$ is served by first-echelon route $i \in \mathcal{T}^1$, 0 otherwise.

$\beta_{ic} = 1$ if customer $c \in C$ is served by second-echelon route $i \in \mathcal{T}^2$, 0 otherwise.

$\epsilon_{ps} = 1$ if satellite $s \in S$ may be served by platform $p \in P$, 0 otherwise;

$\varphi_{sc} = 1$ if customer $c \in C$ may be served by satellite $s \in S$, 0 otherwise.

Decision variables:

$y_i = 1$ if a facility is open at node $i \in P \cup S$, 0 otherwise.

$x_i = 1$ if second-echelon route $i \in \mathcal{T}^2$ is selected, 0 otherwise;

$r_i = 1$ if a first-echelon route $i \in \mathcal{T}^1$ is selected, 0 otherwise;

$f_i \geq 0$ is the flow traveling on first-echelon route $i \in \mathcal{T}^1$ from platforms to satellites;

$$Minimize : \sum_{p \in P} H_p y_p + \sum_{s \in S} H_s y_s + h^1 \sum_{i \in \mathcal{T}^1} r_i + h^2 \sum_{i \in \mathcal{T}^2} x_i + \sum_{i \in \mathcal{T}^1} \pi_i r_i + \sum_{i \in \mathcal{T}^2} \pi_i x_i. \tag{A.1}$$

which are subject to

$$\sum_{s\in S}\varphi_{sc}y_s = 1, \qquad \forall c\in C \tag{A.2}$$

$$\sum_{i\in\mathcal{T}^2}\beta_{ic}x_i = \sum_{j\in S}\varphi_{jc}y_j, \qquad \forall c\in C \tag{A.3}$$

$$\sum_{p\in P}\epsilon_{ps}y_p = y_s, \qquad \forall j\in S \tag{A.4}$$

$$\sum_{i\in\mathcal{T}^1}\alpha_{is}r_i = \sum_{p\in P}\epsilon_{ps}y_p, \qquad \forall s\in S \tag{A.5}$$

$$\sum_{i\in\mathcal{T}^1}f_i\alpha_{is} - \sum_{c\in C}D_c\varphi_{sc}y_s = 0, \qquad \forall p\in P \tag{A.6}$$

$$\sum_{c\in C}D_c\varphi_{sc}y_s \le Q_s y_s, \qquad \forall s\in S \tag{A.7}$$

$$\sum_{i\in\mathcal{T}^1_p}f_i \le Q_p y_s, \qquad \forall p\in P \tag{A.8}$$

$$q^1 r_i - f_i \ge 0, \qquad \forall i\in\mathcal{T}^1 \tag{A.9}$$

$$r_i \in \{0,1\}, \qquad \forall i\in\mathcal{T}^1 \tag{A.10}$$

$$x_i \in \{0,1\}, \qquad \forall i\in\mathcal{T}^2 \tag{A.11}$$

$$y_p \in \{0,1\}, \qquad \forall p\in P \tag{A.12}$$

$$y_s \in \{0,1\}, \qquad \forall s\in S \tag{A.13}$$

$$f_i \ge 0, \qquad \forall s\in\mathcal{T}^1 \tag{A.14}$$

Objective function (A.1) minimizes the total cost. For the second echelon, Constraints (A.2) guarantee that each customer is served by a single satellite while Constraints (A.3) ensure that a customer served by a satellite must be visited by only one vehicle originating from that satellite. Constraints (A.4) and (A.5) ensure the same routing conditions for the first echelon. Constraints (A.6) are flow balance constraints for satellites. Constraints (A.7) and (A.8) enforce capacity restrictions for satellites and platforms, respectively. Constraints (A.9) are consistency constraints between flow and routing variables. Finally, Constraints (A.10)-(A.14) are binary constraints and non-negativity restrictions.

Appendix B.

The Multi-Depot Capacitated Location Routing Problem formulation proposed for the first echelon with multiple platforms comprises of four decision variables:

$x_{ij}^k = 1$ if vehicle k travels from node i to node j, 0 otherwise.
$y_j = 1$ if depot j is open.

$l_{ij}^k \in \mathbb{R}$ corresponds to the load being transported by vehicle k from node i to node j.
$v_k = 1$ if vehicle k is being used, 0 otherwise.

Assume we have a set S^o of open satellites and a demand D'_s for each satellite $s \in S^o$ which equals the sum of the demand of all customers served by s. The following mixed integer programming formulation corresponds to the first-echelon CLRP:

$$min : \sum_{j \in P} H_j y_j + \sum_{k \in K^1} h^1 v_k + \sum_{(i,j) \in E} \sum_{k \in K^1} e_{ij} x_{ij}^k. \tag{B.1}$$

$$\sum_{k \in K^1} \sum_{(j,i) \in E} x_{ji}^k = 1, \qquad \forall i \in S^o \tag{B.2}$$

$$\sum_{j \in P} \sum_{i \in S^o} x_{ji}^k \leq v_k, \qquad \forall k \in K^1 \tag{B.3}$$

$$\sum_{k \in K^1} \sum_{j \in V} (x_{ij}^k - x_{ji}^k) = 0, \qquad \forall i \in V \tag{B.4}$$

$$\sum_{i \in S^o} x_{ji}^k \leq y_j, \qquad \forall j \in P, k \in K^1 \tag{B.5}$$

$$\sum_{j \in V} l_{ij}^k = \sum_{j \in V} (l_{ji}^k - D'_i x_{ij}^k), \qquad \forall i \in S^o, k \in K^1 \tag{B.6}$$

$$l_{ij}^k \leq q^1 x_{ij}^k, \qquad \forall (i,j) \in E, k \in K^1 \tag{B.7}$$

$$\sum_{k \in K^1} \sum_{i \in S^o} l_{ji}^k \leq Q_j y_j, \qquad \forall j \in P \tag{B.8}$$

$$x_{ji}^k \in \{0,1\}, \qquad \forall i,j \in V, k \in K^1 \tag{B.9}$$

$$y_j \in \{0,1\}, \qquad \forall j \in P \tag{B.10}$$

$$v_k \in \{0,1\}, \qquad \forall k \in K^1 \tag{B.11}$$

$$l_{ji}^k \geq 0, \qquad \forall i,j \in V, k \in K^1 \tag{B.12}$$

Objective function (B.1) minimizes the cost of open platforms, employed vehicles and travel time. Constraints (B.2) ensure each satellite is served once and once only by a platform. Constraints (B.3) guarantee each vehicle is used at most once. Constraints (B.4) are flow conservation constraints which ensure that the number of edges entering and leaving a node is equal. Constraints (B.5) guarantee that if an edge is leaving a platform, this platform must be open. Constraints (B.6) concern load conservation and avoid the occurrence of sub-routes by specifying that the load entering node i must be equal to the load leaving node i minus the demand of node i. Finally, Constraints (B.7) and (B.8) are the vehicle and platform capacity constraints.

Appendix C.

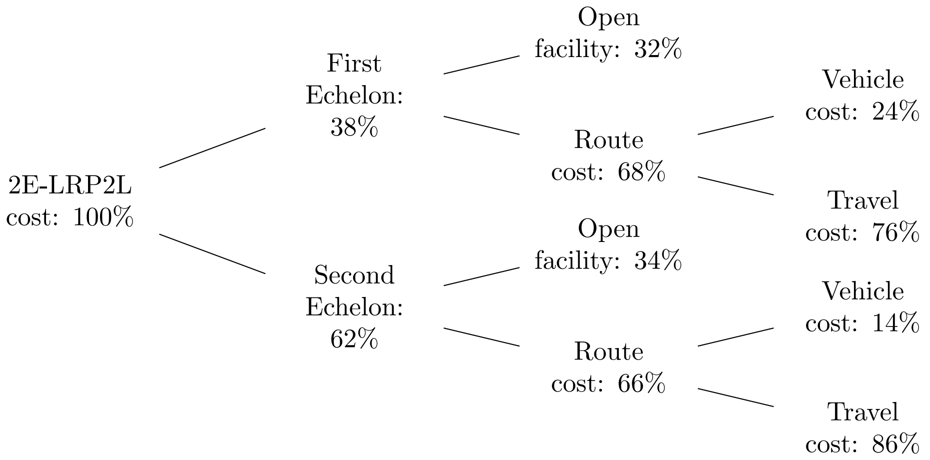


Figure C.11: The average cost breakdown of 2E-LRP2L solutions.

References

Boccia, M., Crainic, T. G., Sforza, A., and Sterle, C. (2010). A metaheuristic for a two echelon location-routing problem. In *International Symposium on Experimental Algorithms*, pages 288–301. Springer.

Boccia, M., Crainic, T. G., Sforza, A., and Sterle, C. (2011). Location-routing models for designing a two-echelon freight distribution system. *Rapport technique, CIRRELT, Université de Montréal*.

Bortfeldt, A. and Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20.

Breunig, U., Schmid, V., Hartl, R. F., and Vidal, T. (2016). A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research*, 76:208–225.

Burke, E. K., Kendall, G., and Whitwell, G. (2004). A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4):655–671.

Christiaens, J. and Vanden Berghe, G. (2020). Slack induction by string removals for vehicle routing problems. *Transportation Science*. doi: 10.1287/trsc.2019.0914.

Clarke, G. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.

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

Côté, J.-F., Gendreau, M., and Potvin, J.-Y. (2014). An exact algorithm for the two-dimensional orthogonal packing problem with unloading constraints. *Operations Research*, 62(5):1126–1141.

Cuda, R., Guastaroba, G., and Speranza, M. G. (2015). A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199.

Drexl, M. and Schneider, M. (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283–308.

Fowler, R. J., Paterson, M. S., and Tanimoto, S. L. (1981). Optimal packing and covering in the plane are np-complete. *Information Processing Letters*, 12(3):133–137.

Fuellerer, G., Doerner, K. F., Hartl, R. F., and Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 36(3):655–673.

Gendreau, M., Iori, M., Laporte, G., and Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3):342–350.

Gendreau, M., Iori, M., Laporte, G., and Martello, S. (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks: An International Journal*, 51(1):4–18.

Imahori, S. and Yagiura, M. (2010). The best-fit heuristic for the rectangular strip packing problem: An efficient implementation and the worst-case approximation ratio. *Computers & Operations Research*, 37(2):325–333.

Iori, M. and Martello, S. (2010). Routing problems with loading constraints. *Top*, 18(1):4–27.

Iori, M., Salazar-González, J.-J., and Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2):253–264.

Jacobsen, S. K. and Madsen, O. B. (1980). A comparative study of heuristics for a two-level routing-location problem. *European Journal of Operational Research*, 5(6):378–387.

Kulkarni, R. and Bhave, P. R. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1):58–67.

Kumar, S. N. and Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 4(03):66.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.

Leung, S. C., Zheng, J., Zhang, D., and Zhou, X. (2010). Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible Services and Manufacturing Journal*, 22(1-2):61–82.

Leung, S. C., Zhou, X., Zhang, D., and Zheng, J. (2011). Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 38(1):205–215.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search. In *Handbook of Metaheuristics*, pages 320–353. Springer.

Nagy, G. and Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672.

Nguyen, V.-P., Prins, C., and Prodhon, C. (2012a). A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 25(1):56–71.

Nguyen, V.-P., Prins, C., and Prodhon, C. (2012b). Solving the two-echelon location routing problem by a grasp reinforced by a learning process and path relinking. *European Journal of Operational Research*, 216(1):113–126.

Passmark (accessed May 11, 2020). *CPU benchmarks*. `http://www.cpubenchmark.net/cpu_list.php`.

Perboli, G., Tadei, R., and Vigo, D. (2011). The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, 45(3):364–380.

Pichka, K., Bajgiran, A. H., Petering, M. E., Jang, J., and Yue, X. (2018). The two echelon open location routing problem: Mathematical model and hybrid heuristic. *Computers & Industrial Engineering*, 121:97–112.

Pirkwieser, S. and Raidl, G. R. (2010). Variable neighborhood search coupled with ilp-based very large neighborhood searches for the (periodic) location-routing problem. In *International Workshop on Hybrid Metaheuristics*, pages 174–189. Springer.

Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., and Limbourg, S. (2015). Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330.

Potvin, J.-Y. and Rousseau, J.-M. (1995). An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446.

Prodhon, C. and Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

Salhi, S. and Nagy, G. (1999). Consistency and robustness in location-routing. *Studies in Locational Analysis*, (13):3–19.

Salhi, S. and Rand, G. K. (1989). The effect of ignoring routes when locating depots. *European Journal of Operational Research*, 39(2):150–156.

Schneider, M. and Drexl, M. (2017). A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1-2):389–414.

Schwengerer, M., Pirkwieser, S., and Raidl, G. R. (2012). A variable neighborhood search approach for the two-echelon location-routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 13–24. Springer.

Vidal, T., Laporte, G., and Matl, P. (2019). A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*.

Wei, L., Zhang, Z., Zhang, D., and 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(3):843–859.

Wei, L., Zhang, Z., Zhang, D., and Lim, A. (2015). A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 243(3):798–814.

Zachariadis, E. E., Tarantilis, C. D., and Kiranoudis, C. T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195(3):729–743.

Zachariadis, E. E., Tarantilis, C. D., and Kiranoudis, C. T. (2013). Integrated distribution and loading planning via a compact metaheuristic algorithm. *European Journal of Operational Research*, 228(1):56–71.