A Simheuristic Approach for the Two-Dimensional Vehicle Routing Problem with Stochastic Travel Times

Daniel Guimarans[a], Oscar Dominguez[b], Javier Panadero[c,*], Angel A. Juan[c]

[a] *Aviation Academy*
Amsterdam University of Applied Sciences, Amsterdam, Netherlands
d.guimarans.serrano@hva.nl
[b] *Opein Inc., Gran Canaria, Canary Islands*
oscar@opein.com
[c] *IN3 – Computer Science, Multimedia and Telecommunication Dept.*
Universitat Oberta de Catalunya, Barcelona, Spain
{jpanaderom, ajuanp}@uoc.edu

Abstract

The two-dimensional vehicle routing problem (2L-VRP) is a realistic extension of the classical vehicle routing problem in which customers' demands are composed by sets of non-stackable items. Examples can be found in real-life applications such as the transportation of furniture or industrial machinery. Often, it is necessary to consider stochastic travel times due to traffic conditions or customers availability. However, there is a lack of works discussing stochastic versions of the 2L-VRP. This paper offers a model of the 2L-VRP with stochastic travel times that also includes penalty costs generated by overtime. To solve this stochastic and non-smooth version of the 2L-VRP, a hybrid simheuristic algorithm is proposed. Our approach combines Monte Carlo simulation, an iterated local search framework, and biased-randomised routing and packing heuristics. Our algorithm is tested on an extensive benchmark, which extends the deterministic one for the 2L-VRP with unrestricted and non-oriented loading.

Keywords: Simulation-Optimisation, Simheuristics, Biased-Randomised Heuristics, Transportation, Vehicle Routing Problem, Packing Problem.

*Corresponding author

1. Introduction

The vehicle routing problem (VRP) is a well-known combinatorial optimisation problem in which a fleet of vehicles has to service a set of customers at the lowest possible cost (Golden et al., 2008; Toth and Vigo, 2014). The most basic variant of the VRP is the capacitated vehicle routing problem (CVRP), where a homogeneous fleet of vehicles with restricted capacity, based at a central depot, needs to satisfy customers demands by visiting them only once. Additional restrictions, such as distance or time-based constraints, are often considered in richer variants of the problem. Many VRP variants have been extensively studied due to their challenging *NP-hard* nature and their potential applicability on real-life transportation activities (Caceres-Cruz et al., 2014; Lahyani et al., 2015). In this work, we consider a realistic variant of the CVRP that combines vehicle routing and loading (packing) constraints, known as two-dimensional VRP (2L-VRP) (Iori et al., 2007). In a 2L-VRP, customers demands consist of a set of rectangular items that cannot be stacked due, for instance, to their weight or fragility. Figure 1 shows an example of a 2L-VRP with three routes and their corresponding packing plans. Our work was originally motivated by operations at Opein Inc. (www.opein.com), a medium-size company in Spain that hires industrial equipment to its customers, mostly in the construction field. Opein has to periodically deliver and pick up a variety of industrial machinery (e.g., aerial-work platforms, energy-generation sets, dumpers, forklifts, or professional cleaning equipment). Similar issues are also present in other transportation activities, where large-size items – such as furniture or appliances – are required to be picked up or delivered. These items must be efficiently accommodated in the truck to ensure a high utilisation of the vehicle. This packing process may also critically affect routing decisions. Hence, one needs to consider not only the items weight, but also their dimensions. In the 2L-VRP, items are normally considered to be of rectangular shape and they cannot be piled up or overlap on the truck's loading surface.

Four main variants have been defined for the 2L-VRP, depending on different configurations of the loading constraints. These constraints may refer to the order items are loaded on the vehicle (unrestricted or sequentially) or the orientation of each item (allowing, or not, items rotation). The different combinations of these constraints yield the four 2L-VRP variants: *(i)* unrestricted oriented loading ($2|UO|L$), allowing items to be rearranged during the distribution process but not rotated; *(ii)* unrestricted non-oriented

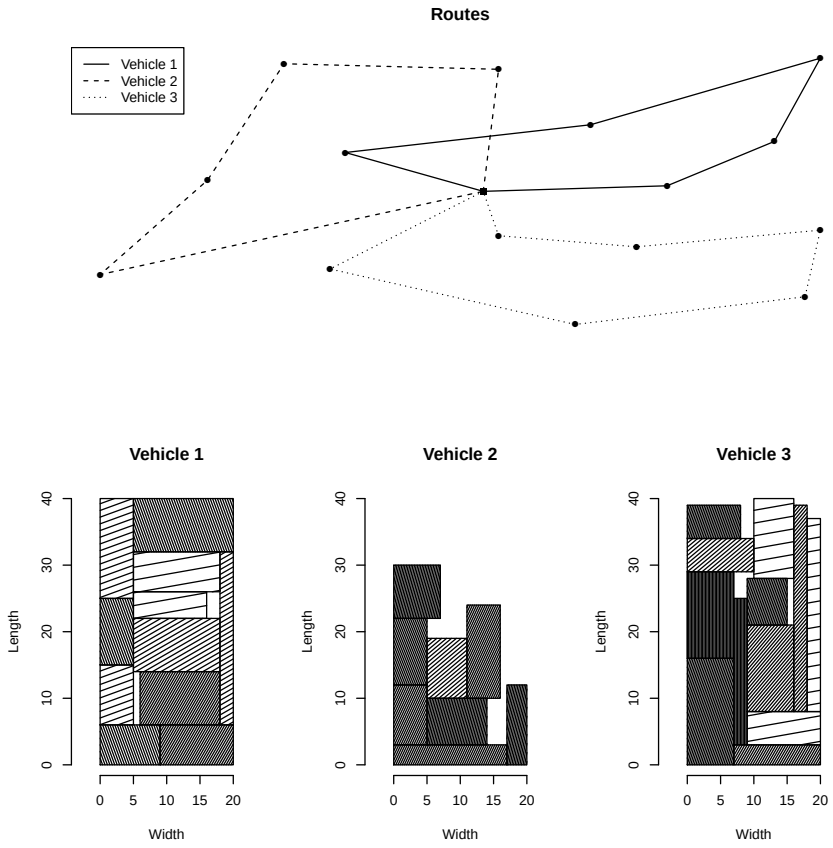


Figure 1: Example of a set of routes with their corresponding packing plans.

(rotated) loading ($2|UR|L$), where items can be rearranged and rotated to better fit on the loading surface; *(iii)* sequential oriented loading ($2|SO|L$), restricting items to be loaded without rotation in reverse order to customer visits; and *(iv)* sequential non-oriented (rotated) loading ($2|SR|L$), allowing items to be rotated, but ensuring they are loaded in reverse order to the visits in the route. In this work, we tackle the unrestricted non-oriented version of the problem ($2|UR|L$), although the methodology proposed here can be extended to cope with other variants and extensions of the 2L-VRP. Allowing items rotation is not only a realistic assumption, but it also may yield more efficient packing plans (Dominguez et al., 2014b).

One of the main reported issues in real operations is the significant vari-

ability on the time required to perform all the assigned activities. Although routes may be planned carefully, travel times between customers depend on traffic and access conditions to the site, causing the actual time to vary substantially with respect to the original estimate. Likewise, customers availability may also delay the start of the service, forcing the vehicle to wait at the site upon arrival. Since all these potential sources of delay occur between the end of the previous service and the start of the next one, they are generally aggregated as deviations from the planned travel time. These deviations from the original plan translate into an increase of operational costs. Often, drivers need to be paid overtime due to an excess of driving hours beyond the duration of their shift. Furthermore, the company may incur into additional costs related to extending their operation time, like paperwork needs to be finished after the completion of all routes. Deterministic solving approaches for the 2L-VRP fail to capture such real-life uncertainty. Accordingly, the main contributions of this paper are: *(i)* the introduction of a more realistic version of the 2L-VRP that considers stochastic travel times (2L-SVRP) and penalty costs due to overtime; *(ii)* the proposal of a formal model to describe the 2L-SVRP in an accurate way, including the use of penalty-cost functions to account for overtime; *(iii)* the development of a simheuristic algorithm (Juan et al., 2015), integrating simulation within a metaheuristic framework, to minimise the expected total cost that includes both travel-times cost and overtime cost; and *(iv)* the generation of a new set of stochastic instances for the 2L-SVRP by extending, in a natural way, the classical sets for the 2L-VRP. Our simheuristic approach relies on an iterated local search (ILS) framework (Lourenço et al., 2010; Grasas et al., 2016a), where we embed biased-randomised versions of classical routing and packing heuristics. Biased randomisation of heuristics refers to the use of skewed probability distributions to induce an oriented (biased) random behaviour of the heuristic, transforming a deterministic method into a probabilistic algorithm (Juan et al., 2013a; Grasas et al., 2017). In particular, we use a biased-randomised version of the classical Clarke and Wright savings heuristic for the CVRP (Clarke and Wright, 1964), enhanced with memory-based techniques (Juan et al., 2010). To compute packing plans, we use a biased-randomised version of the Best-Fit heuristic (Burke et al., 2004). Monte Carlo simulation (MCS) is integrated at two stages of the search process in order to: *(i)* assess the performance of the obtained solutions; and *(ii)* to provide guidance to the metaheuristic component during the solution-searching process. Our method is able to obtain solutions that clearly outperform the

ones obtained for the deterministic version of the problem when these are applied in a stochastic environment. In other words, near-optimal solutions for the deterministic 2L-VRP might easily become sub-optimal solutions for the 2L-SVRP, which justifies the need of simulation-based methods as the one introduced here. Additionally, our simheuristic approach could easily be adapted so that, instead of searching for the solution with the minimum expected cost, it searches for the solution that maximises the probability of not exceeding a specific cost threshold. Notice that the use of more traditional methods, such as stochastic programming, would derive in prohibitive computing times when solving medium- and large-size instances of the 2L-SVRP, specially considering that the introduction of penalty costs makes the objective function to become non-smooth.

The remaining of this paper is structured as follows. We provide an overview on related work in Section 2 and formalise the 2L-SVRP problem in Section 3. Details of our proposed approach are described in Section 4. Our simheuristic algorithm is assessed on a set of numerical experiments, explained and discussed in Section 5. Finally, Section 6 summarises the main contributions and results of this work.

2. Literature Review

To the best of our knowledge, no other authors have published analysis on the 2L-SVRP. For that reason, we have distributed the literature review in three parts. First, a review on existing works on the deterministic 2L-VRP is provided. Then, we discuss existing work on stochastic variants of the VRP. Finally, a review is given on works using simheuristic approaches for solving stochastic optimisation problems.

2.1. The Deterministic 2L-VRP.

The deterministic version of the 2L-VRP is a relatively recent variant of the VRP, originally introduced by Iori et al. (2007). It is another example of a research trend aimed at including more real-life constraints into classical VRP variants (Caceres-Cruz et al., 2014; Lahyani et al., 2015). Specific surveys on the combination of routing and loading constraints can be found in Wang et al. (2009) and Iori and Martello (2010).

In the original work by Iori et al. (2007), the authors propose an exact branch-and-cut algorithm to solve the routing aspects, and a branch-and-bound algorithm combined with heuristics and effective lower bounds to deal

with the packing requirements. From the different loading configurations, the authors only tackle the sequential oriented case.

Although the approach by Iori et al. (2007) is able to find optimal solutions for instances up to 35 customers and 100 items, there are smaller instances where optimality has not been guaranteed (Iori and Martello, 2010). Since these limits are not reasonable for practical applications, several approaches based on heuristics and metaheuristics have been proposed in the literature. Different authors have developed approaches based on tabu search (TS) algorithms. Thus, Gendreau et al. (2008) propose a TS approach designed to solve both the sequential and the unrestricted oriented loading configurations. Zachariadis et al. (2009) use a hybrid algorithm combining TS, guided local search, and five packing heuristics with different selection criteria to develop feasible loading configurations. Leung et al. (2011) combine TS with a new heuristic for solving the loading configuration. The same authors had previously presented a simulated annealing (SA) approach to tackle the oriented versions of the problem (Leung et al., 2010). Duhamel et al. (2009) propose a hybrid approach combining GRASP (Greedy Randomised Adaptive Search Procedure) with evolutionary local search. In their approach, the loading constraint is handled by a resource constraint project scheduling problem heuristic especially tuned to address the bin packing part of the problem. The same authors later embedded their approach into an optimisation framework whose last step was transforming the obtained relaxed solutions into 2L-VRP solutions by solving a dedicated packing problem (Duhamel et al., 2011). Zachariadis et al. (2013) present an approach named Promise Routing-Memory Packing (PRMP), which combines local search with an effective diversification based on regional aspiration criteria. The loading feasibility of routes is tackled by a packing heuristic enhanced by an innovative memory mechanism. The same authors have extended their approach to tackle the 2L-VRP with unrestricted oriented and non-oriented loading, as well as a new variant of the problem with simultaneous pick-ups and deliveries (Zachariadis et al., 2016, 2017). Finally, Wei et al. (2015) introduced a variable neighbourhood search (VNS) algorithm to tackle both sequential and unrestricted oriented loading variants of the 2L-VRP. The authors combine a skyline heuristic with memory-based mechanisms to examine loading constraints. More recently, Wei et al. (2018a) also combined an open space-based heuristic and efficient data structures within a SA framework to face the unrestricted and sequential variants of the problem, both with and without items rotation.

The oriented versions of the problem have received far more attention in the 2L-VRP literature than the non-oriented variants. Fuellerer et al. (2009) were the first authors to address the four problem variants, using an ant colony approach. In fact, only Fuellerer et al. (2009), Dominguez et al. (2014b; 2016a), Zachariadis et al. (2016; 2017), and Wei et al. (2018a) have addressed the non-oriented loading configurations. Dominguez et al. (2014b) propose a multi-start biased-randomised algorithm to analyse both unrestricted oriented and non-oriented loading variants. At each restart, a biased randomisation of a savings-based heuristic is combined with an enhanced version of a classical packing heuristic to produce feasible good solutions. A similar approach was used later to solve the sequential oriented and non-oriented variants of the problem, as well as a 2L-VRP extension with backhauling (Dominguez et al., 2016a). A limited number of problem extensions have also been addressed in the literature, both with oriented and non-oriented loading configurations, including variants with heterogeneous fleet (Dominguez et al., 2014a, 2016b), backhauls (Dominguez et al., 2016a; Zachariadis et al., 2017), or simultaneous pick-ups and deliveries (Zachariadis et al., 2016, 2017).

2.2. Stochastic Variants of the VRP.

None of the aforementioned 2L-VRP extensions considers the time variability associated with real-life operations. However, VRP variants considering stochastic travel times have been previously studied in the literature, although to a limited extent. Most approaches focus on uncapacitated versions of the problem, therefore effectively addressing a variant of the multiple travelling salesman problem (mTSP). Laporte et al. (1992) studied this problem including a penalty for excess time incurred in a route. The authors formulated three mathematical models and used a branch-and-cut approach capable of solving to optimality small instances. Lambert et al. (1993) presented an extension of this work considering hard deadlines and developed a savings-based heuristic procedure to solve this variant. These authors used this approach to solve instances derived from real scenarios of the branch network of a Belgian bank. Kenyon and Morton (2003) also address the mTSP with stochastic travel times and consider two models with different objectives: minimising the expected completion time of all routes, and maximising the probability that operations are completed before a defined target time. They use a branch-and-cut approach combined with MCS to solve instances of modest size. To the best of our knowledge, Rostami et al. (2017)

were the first authors to address the CVRP with stochastic and correlated travel times. They used a mean-variance approach to solve instances of up to 75 customers. Only a previous work by Lecluyse et al. (2009) had addressed a related CVRP with time-dependent travel times using a TS approach. Other VRP variants considering time windows and stochastic service times have also been studied in the literature (Zhang et al., 2013; Errico et al., 2016a). This variant is often used to represent stochastic demands in versions of the VRP present in service industries (e.g., technicians, healthcare, etc.). We refer the reader interested on this problem variant to the survey by Errico et al. (2016b).

2.3. Applications of the Simheuristics Methodology.

As far as we know, only a preliminary work by Guimarans et al. (2016) tackles a stochastic variant of the 2L-VRP. These authors present an approach based on a multi-start strategy, although providing exploratory results. Our simheuristic algorithm significantly extends their previous work by introducing a more complete and efficient ILS-based solving methodology, as well as a more extensive set of numerical results.

Simheuristics have proved to be an efficient approach to solve stochastic problems (Juan et al., 2015). They combine metaheuristic methods with simulation to guide the algorithm towards solutions that may behave better in variable environments (e.g., less sensitive to plan deviations or better performance in worst-case scenarios). The extension of traditional metaheuristics into simheuristics is gaining popularity (Grasas et al., 2016b), and they have been applied already to solve stochastic optimisation problems in fields such as road transportation (Juan et al., 2011b, 2013b; Gonzalez-Martin et al., 2018), waste collection (Gruler et al., 2017a,b), aviation (Guimarans et al., 2015), inventory routing (Juan et al., 2014b; Gruler et al., 2018a,b), scheduling (Juan et al., 2014a; Gonzalez-Neira et al., 2017; Hatami et al., 2018), facility location (de Armas et al., 2016), or finance (Panadero et al., 2018).

3. Problem Definition

In the 2L-SVRP, a complete undirected graph $G = (V, E)$ is given, where $V = \{0, 1, 2, \ldots, n\}$ is the set of nodes representing the depot (node 0) and the n customers (V_0), while $E = \{(i, j) \mid i, j \in V;\ i \neq j\}$ is the set of edges connecting these locations. The edges set $E = \{(i, j) \mid i, j \in V;\ i \neq j\}$ connects customers with each other and with the depot. In the best case scenario

(i.e., with no traffic jams, etc.), traversing edge (i, j) takes a minimum travel time $t_{ij} = t_{ji} > 0$. However, under uncertainty conditions, the travel time necessary to traverse the edge becomes a random variable $T_{ij} = t_{ij} + D_{ij}$, where $D_{ij} = D_{ji} \geq 0$ is a random delay that follows a known non-negative probability distribution (e.g., Log-Normal, Weibull, etc.). Transportation of goods is performed by a fleet of $K > 1$ identical vehicles, initially located at the depot, each with maximum weight-loading capacity $Q > 0$ and a loading area $A = W \times H$. For each customer $i \in V_0$, there are $m_i > 0$ items with a total weight $q_i > 0$ to be delivered. It is assumed that the depot has no demand, i.e., $m_0 = 0$. For each customer's item, its length and width dimensions are denoted by $h_{il} > 0$ and $w_{il} > 0$ $(1 \leq l \leq m_i)$, respectively. Thus, the total area covered by the items of customer i can be denoted as $a_i = \sum_{l=1}^{m_i} w_{il} h_{il}$. The variable $z_{ijk} \in \{0, 1\}$ is used for routing decisions, such that an edge (i, j) is part of the solution if, and only if, z_{ijk} takes the value 1 for vehicle $k \in K$, while it takes the value 0 otherwise.

The resolution of the 2L-SVRP consists in finding a set of routes that minimise the expected time-based cost, such that: *(i)* every route starts and ends at the depot; *(ii)* each customer is visited exactly once (therefore, all its demanded items should fit into a single vehicle); and, *(iii)* in each route, customers items do not exceed vehicles capacity and loading surface.

The routing aspects of the problem can be formulated as follows:

$$\min \mathbb{E}\left[\sum_{k \in K} \sum_{\substack{i,j \in V \\ i \neq j}} T_{ij} z_{ijk}\right] = \sum_{k \in K} \sum_{\substack{i,j \in V \\ i \neq j}} \mathbb{E}\left[T_{ij}\right] z_{ijk} \tag{1}$$

subject to:

$$\sum_{j \in V_0} z_{0jk} = \sum_{i \in V_0} z_{i0k} \quad \forall k \in K \tag{2}$$

$$\sum_{k \in K} \sum_{i \in V} z_{ijk} = 1 \quad \forall j \in V_0 \tag{3}$$

$$\sum_{i \in V} z_{iuk} = \sum_{j \in V} z_{ujk} \quad \forall u \in V_0, \forall k \in K \tag{4}$$

$$\sum_{k \in K} \sum_{j \in V_0} z_{0jk} \leq K \tag{5}$$

$$\sum_{\substack{i,j \in V \\ i \neq j}} q_i z_{ijk} \leq Q \quad \forall k \in K \tag{6}$$

$$\sum_{\substack{i,j \in V \\ i \neq j}} a_i z_{ijk} \leq A \quad \forall k \in K \tag{7}$$

$$z_{ijk} \in \{0,1\} \quad \forall i,j \in V, i \neq j, \forall k \in K \tag{8}$$

The objective function (1) minimises the expected cost (travel time in this case) required to service all customers. Constraint (2) ensures that the number of vehicles leaving the depot is the same as the number of vehicles returning to it. Equations (3) and (4) enforce that each customer is visited exactly once and that if a vehicle visits a customer, it must also depart from it. Constraint (5) ensures that the maximum number of vehicles is never exceeded. Finally, Equations (6) and (7) impose that the maximum capacity and loading area of the vehicle are respected, respectively.

In addition, we need to take into account packing feasibility in every route. In order to regard a packing configuration as feasible, all items must be loaded without overlapping and with the edges parallel to the edges of the vehicle (orthogonal loading). In the $2|UR|L$ case, items may be rotated when loaded into the vehicle. Hence, we start by defining the loading surface of a vehicle as a $W \times H$ matrix with indexes $x \in \{1,2,\ldots,W\}$ and $y \in \{1,2,\ldots,H\}$. Hence, the position of the bottom-left corner of item l from customer i, loaded on vehicle k, is denoted by coordinates (x_{ilk}, y_{ilk}). We also define a route R as a subset of customers forming a route or partial route ($R \subseteq V_0$). Finally, we define the variable Ω_{il} to indicate if item l of customer i has been rotated ($\Omega_{il} = 1$) or not ($\Omega_{il} = 0$). Using this notation, we can define the packing constraints as follows:

$$\begin{aligned} &0 \leq x_{il} \leq (W - w_{il})(1 - \Omega_{il}) + (W - h_{il})\Omega_{il} \quad \wedge \\ &0 \leq y_{il} \leq (H - h_{il})(1 - \Omega_{il}) + (H - w_{il})\Omega_{il} \\ &\forall i \in R, \forall l \in \{1,2,\ldots,m_i\} \end{aligned} \tag{9}$$

$$
\begin{aligned}
& x_{il} + w_{il}(1 - \Omega_{il}) + h_{il}\Omega_{il} \leq x_{jl'} && \vee \\
& x_{jl'} + w_{jl'}(1 - \Omega_{jl'}) + h_{jl'}\Omega_{jl'} \leq x_{il} && \vee \\
& y_{il} + h_{il}(1 - \Omega_{il}) + w_{il}\Omega_{il} \leq y_{jl'} && \vee \\
& y_{jl'} + h_{jl'}(1 - \Omega_{jl'}) + w_{jl'}\Omega_{jl'} \leq y_{il} && \\
& \forall i, j \in R, \forall l \in \{1, 2, \ldots, m_i\}, \forall l' \in \{1, 2, \ldots, m_j\}, (i, l) \neq (j, l') && (10)
\end{aligned}
$$

Constraint (9) ensures that items are loaded within the vehicle's loading surface, while expression (10) permits avoiding any two items overlapping on the surface of the vehicle.

As mentioned in Section 1, companies such as Opein Inc. often have to deal with additional costs due to overtime and associated expenses. This problem is generally aggravated by unforeseen deviations from planned operations, normally due to longer-than-expected travel or service times. In order to adjust the 2L-SVRP formulation to these situations, we have modified the objective function (1) to account for incurred further expenses. Thus, whenever the travel time employed by one vehicle exceeds a user-given threshold (e.g., 8 hours in some real-life cases), a penalty cost is added to the objective function. Hence, the new objective function can be expressed as (11):

$$
\min \mathbb{E}\left[\sum_{k \in K} f_k\right] \tag{11}
$$

with:

$$
f_k = \begin{cases} \sum\limits_{\substack{i,j \in V \\ i \neq j}} T_{ij} z_{ijk} & \text{, if } \sum\limits_{\substack{i,j \in V \\ i \neq j}} T_{ij} z_{ijk} \leq \tau \\ \sum\limits_{\substack{i,j \in V \\ i \neq j}} T_{ij} z_{ijk} + \rho(\sum\limits_{\substack{i,j \in V \\ i \neq j}} T_{ij} z_{ijk} - \tau) + \gamma & \text{, otherwise} \end{cases} \tag{12}
$$

where $\rho > 0$ and $\gamma \geq 0$ are user-given parameters associated with a penalty cost component due to the existence of overtime in completing route k. Notice that we are assuming that this penalty cost can be expressed in time units (or, equivalently, that the travel time cost can be expressed in monetary units).

4. Our Simheuristic Solving Approach

We propose a simheuristic approach for minimising the expected (time-based) total cost in the 2L-SVRP. Our methodology combines an ILS metaheuristic with simulation techniques to deal with the stochastic nature of the problem. As discussed in Grasas et al. (2016a), ILS offers a well-balanced combination of efficiency and relative simplicity, and can be easily extended to a simheuristic.

Algorithm 1 depicts the main characteristics of our approach, composed of three stages. First, a feasible initial solution is generated using a constructive heuristic. Then, during the second stage, the ILS metaheuristic improves this initial solution by iteratively exploring the search space and conducting a short number of MCS runs. This procedure is based on perturbing the current solution to obtain a new starting point, and subsequently exploring the neighbourhood of this new solution. The short MCS runs are used to obtain rough estimates of the solution's behaviour under stochastic conditions, which allows to generate a pool of 'promising' elite solutions. In the third stage, a refinement procedure using a larger number of MCS runs is applied to these elite solutions, which allows to obtain a more accurate estimation of the expected total cost. The following subsections describe in more detail each step of our approach.

Algorithm 1 ILS-based Simheuristic (*inputs*, α, β, $maxPackIter$, K_{min}, K_{max}, λ, T_0, t_{max})

```
1: initSol ← genInitSol(inputs, α, β, maxPackIter ) % Initial solution stage
2: baseSol ← initSol
3: bestSol ← baseSol
4: fastSimulation(baseSol) % Monte Carlo simulation
5: T ← T_0
6: while (time ≤ t_max) do % ILS stage
7:     k ← K_min
8:     while (k ≤ K_max) do
9:         newSol ← shaking(baseSol, k, α, β, maxPackIter) % BR heuristic
10:        newSol ← localSearch(newSol)
11:        if (detCost(newSol) < detCost(baseSol)) then
12:            fastSimulation(newSol) % Monte Carlo simulation
13:            if (stochCost(newSol) < stochCost(baseSol)) then
14:                baseSol ← newSol
15:                if (stochCost(newSol) < stochCost(bestSol)) then
16:                    bestSol ← newSol
17:                    insert(poolBestSol,bestSol)
18:                end if
19:                k ← K_min
20:            end if
21:        else % SA-based acceptance criterion
22:            temperature ← calcTemperature(detCost(newSol), detCost(baseSol), T)
23:            if (U(0,1) ≤ temperature) then
24:                baseSol ← newSol
25:                k ← K_min
26:            else
27:                k ← k + 1
28:            end if
29:        end if
30:        T ← λT
31:    end while
32: end while
33: for (sol ∈ poolBestSol) do % Refinement stage - Monte Carlo simulation
34:    deepSimulation(sol)
35:    if (stochCost(sol) < stochCost(bestSol)) then
36:        bestSol ← sol
37:    end if
38: end for
39: return bestSol
```

4.1. Construction of the Initial Solution.

With the aim of generating an initial solution, we use a constructive heuristic which employs biased randomisation techniques. As discussed in Juan et al. (2013a) and Grasas et al. (2017), biased-randomised techniques refer to the use of skewed probability distributions to induce an 'oriented' (non-uniform) random behaviour. This process permits turning a deterministic heuristic into a probabilistic algorithm, while still preserving the logic behind the heuristic. In our approach, biased-randomised techniques are applied on the routing and packing heuristics. For the routing component, we use the modified version of the well-known savings heuristic described in (Juan et al., 2011b). As for solving the packing, we use the biased-randomised version of the Best-Fit heuristic described in Dominguez et al. (2016a).

The classical savings heuristic (Clarke and Wright, 1964) starts by generating an initial dummy solution. This dummy solution consists of a return trip from the depot to each customer, using as many vehicles as the number of customers in the problem. Next, we compute the savings associated with each edge, i.e., the cost reduction of including an edge connecting two customers in one route instead of visiting them in two separate routes. These savings are then sorted in descending order. At this point, an iterative process is initiated. In each iteration, the elements in the savings list are rearranged by applying a biased-randomised process, so edges associated with higher savings are more likely to be ranked at the top of the list. This process allows edges to be selected in a different order at each iteration of the process, while still preserving the flavour of the original heuristic. In our case, a geometric probability distribution, driven by a single parameter α ($0 < \alpha < 1$), is used to induce this skewed behaviour. The value for this parameter was set after a quick tuning process over a random sample of deterministic 2L-VRP benchmark instances, establishing a good performance whenever α falls between $(0.3, 0.4)$ (i.e., any random value inside this interval will generate similar results). Once the savings list is computed, the algorithm starts a route-merging process until the list is empty. At each iteration, the edge at the top of the list is selected. This edge connects two routes, which will be merged if: *(i)* there is enough weight capacity in the vehicle to carry all items from both routes; *(ii)* there is enough available surface to load all items from both routes; and *(iii)* there exists a feasible packing for all items such that they can be conveniently loaded without overlapping.

In order to evaluate packing feasibility, a biased-randomised version of the Best-Fit heuristic (Burke et al., 2004) is employed. This is a constructive and

deterministic procedure that selects the next item (rectangle) to pack in the vehicle based on bottom-left and 'best fit' criteria, i.e., among the available items, it always chooses the one that offers the 'best fit' when positioned at the bottom-left of the lowest available space. This deterministic heuristic is transformed into a biased-randomised procedure by assigning probabilities to the different items, so that the better the fit the higher its probability of being chosen. Again, a geometric distribution is introduced to skew the selection, controlled by a single parameter β ($0 < \beta < 1$). Following a tuning process similar to the one used for the routing process, a good performance was observed when β fell in the range $(0.06, 0.23)$. Thus, the value of β was established to be a random number within this interval. The biased-randomised version of the Best-Fit heuristic is encapsulated within a multi-start process. This allows to run the biased-randomised heuristic several times, thus increasing our chances of finding a feasible packing solution. This process is controlled by the parameter $maxPackIter$, which is set to be proportional to the instance size. This multi-start process is combined with the use of a fast-access memory-based method (a *cache* based on a hash map data structure) to speed up the method. We first check if the algorithm has already computed a feasible packing solution for the same configuration (i.e., same combination of customers). If a feasible solution is already known, routes are merged without launching the process. Otherwise, the biased-randomised Best-Fit procedure is started, updating the packing cache memory whenever a feasible solution is found. In both cases, the packing process is stopped, deeming the route merging as feasible, and the route construction resumes.

Note that, while other approaches propose a two-stage method – one for solving the packing problem and another for solving the routing component –, our approach integrates the packing problem as part of the route construction method. This way, we can guarantee the feasibility of all routes, as we explicitly take into account the loading constraints during the construction phase.

4.2. SimILS Main Stage.

The second phase of our approach combines an ILS framework with simulation to assess the behaviour of obtained solutions in stochastic scenarios. ILS is a relatively simple methodology that sequentially applies a diversification method (*shaking*) and local search around the *shaken* solution (Lourenço

et al., 2010). The process is repeated until the stopping criterion is met, normally based on maximum execution time (t_{max}) or number of iterations. Algorithm 1 outlines the main steps of our approach. We start the process by perturbing the current solution (step 9). This process is dependent on the value of k, representing the degree of destruction to be applied in the shaking phase. This value is updated between K_{min} and K_{max} in a VNS fashion, i.e., it is reset whenever a best solution is found (steps 7, 19 and 25) and increased when the algorithm fails to improve the current solution (step 27). During the shaking process, k adjacent routes are selected at random from the current solution, and their corresponding customers are unassigned. Next, in order to complete this partial solution, we apply the constructive biased-randomised versions of the savings and Best-Fit heuristics used to obtain the initial solution. In this process, biased randomisation prevents the same solution from being obtained at every iteration. At the same time, using the same biased-randomised procedures ensures the shaken solution is not far from the original solution, even for high values of k. This characteristic is desirable, given the empirical principle by which good solutions tend to be clustered in the solution space.

Following the shaking process, the algorithm starts a local search around the perturbed solution. This stage consists of two steps. First, a traditional *2-opt* move is used to explore neighbouring solutions. This operator is applied to each route until it cannot be further improved, before moving to the next route. Notice that, since only intra-route movements are evaluated, we do not need to check the packing feasibility of the new sequence. Only in the sequential versions of the 2L-VRP, feasibility might be altered by intra-route moves. In the second step, once the solution cannot be further improved with the *2-opt* operator, memory-based techniques are used to achieve a faster convergence. This mechanism is implemented using a hash map data structure allowing quick access to its elements. As in the packing case, previously-computed routes and packing plans for a given set of customers are stored in this fast-access cache memory. In this case, we keep in memory the best solution found so far for a specific set of customers to be visited. If the obtained solution contains the same set of visits with a higher cost, the route stored in the cache memory is retrieved and the current solution is updated. Otherwise, we add (or update) the route to the cache memory for subsequent iterations.

As mentioned in Section 1, carriers often have to deal with unforeseen costs related to overtime and associated expenses, e.g., due to longer-than-

expected travel times. These additional costs are modelled in our problem by means of a non-smooth objective function given by Equations (11) and (12). In these situations, a solution less sensitive to deviations from the original plan is desirable. So far, the computed solution responds to the deterministic 2L-VRP and does not take into account these potential deviations. In order to deal with the stochastic nature of the proposed problem, we include two simulation processes in our algorithm. In the first case, we run a short MCS accounting for travel time variability whenever the deterministic cost of the base solution is improved (step 12). If the new solution is also able to improve the stochastic cost of the base solution, the latter is updated (step 14). In the same way, if the stochastic cost of the new solution improves the cost of the best solution found so far, the latter is updated (step 16) and added to the pool of elite solutions (step 17). By limiting the size of this pool, we ensure we only keep track of the 'elite' solutions as the algorithm evolves. Notice that the MCS does not only provides estimates to the expected cost associated with the solutions generated by the approach, but it also reports feedback to the stochastic search process. Indeed, the selection of the current base and best solutions is driven by the results of the simulation. In order to further diversify the search, the algorithm might occasionally accept non-improving solutions following an acceptance criterion based on simulated annealing (step 21), with decaying probability regulated with a temperature parameter (T) adjusted at each iteration (step 30).

Once the algorithm is finished, a longer MCS is launched to better assess the elite solutions in the pool (step 34) before reporting the final results. Since the number of generated solutions during the search can be large and the simulation process is time-consuming, we limit the number of MCS iterations to be executed. For our approach, the number of iterations for the exploratory and intensive MCS stages were set to 1,000 and 50,000, respectively.

In both simulation processes, the delay times between any two customers, D_{ij}, are modelled using a Log-Normal probability distribution. As discussed by Juan et al. (2011a), the Log-Normal distribution is quite convenient to model positive random variables such as this one. Since we defined $T_{ij} = t_{ij} + D_{ij}$, the minimum travel time t_{ij} is considered as a lower bound for the random time T_{ij} requested to travel from customer i to customer j. In our experiments, we have set the LogNormal parameters as follows: location parameter $\mu = 0$, and scale parameter $\sigma = 0.75$. An important advantage of our approach is its relative simplicity: our method uses few

parameters, significantly reducing its sensitivity to the particular characteristics of a problem instance.In fact, only parameters $maxPackIter$, K_{min}, K_{max}, and execution time (t_{max}), are adjusted according to the instance size, while other parameters remain fixed. Moreover, all values were established after a quick and simple tuning process, in which different combinations of values were tested for a random sample of instances.

5. Computational Experiments

The proposed algorithm was implemented using the Java programming language. All experiments were run in a standard computer with an Intel Core i7 processor at 2.9 GHz and 8 GB RAM. The algorithm was executed on the Eclipse platform for Java over OS X 10.11. The classical 2L-VRP benchmark set proposed by Iori et al. (2007) and Gendreau et al. (2008) was used in the experimentation. This set consists of 180 instances, divided in 5 classes with 36 instances each. The instances are identified following the nomenclature *Exxx-yyyz*, where *xxx* is the total number of customers, *yyy* is the configuration of the instance, and z identifies the class of the instance – see Toth and Vigo (2014) for the details of each configuration and its notation. Among these classes, instances in class 1 correspond to a more simplified version of the problem, where all customers demand one single item of dimensions 1×1. Therefore, the packing influence is relatively limited during the route construction and this class has not as much interest as the remaining ones.

Thus, we extended all deterministic benchmark instances in classes 2 to 5 to the stochastic case in order to assess our methodology. As mentioned in Section 4, a Log-Normal distribution was selected to model the random delay in travel time between each pair of customers, using the original deterministic value provided in each instance as the minimum travel time required in an ideal scenario without uncertainty. With this extension, it is possible to account for the significant variability in travel times present in real systems – including also potential delays in service times if these are considered as part of the travel times. For our experiments, the remaining parameters were set as described in Section 4 and the algorithm's execution time was adjusted to be proportional to the instance size. Tables 1 to 4 present the results for all the stochastic instances.

These stochastic instances include a time threshold (τ) for each route, which is adjusted for every instance according to the average length of routes

Table 1: Results for Class 2 benchmark instances considering deterministic and stochastic travel times.

			Without Penalties			With Penalties								
#	Instance	τ	BKS [1]	OBS [2]	Δ (%) [2-1]	OBD_{det} [3]	OBS_{det} [4]	OBD_{stoch} [5]	OBS_{stoch} [6]	Δ (%) [4-3]	Δ (%) [6-5]	Δt_{OBS} [7]	CPU_{OBD} (s) [8]	CPU_{OBS} (s) [9]
1	E016-03m2	70	278.73	278.73	0.00	457.62	474.61	553.40	546.23	3.71	-1.30	19.20	0	0
2	E016-05m2	65	334.96	334.96	0.00	578.96	580.34	628.67	584.75	0.24	-6.99	6.75	2	4
3	E021-04m2	70	380.35	380.35	0.00	675.24	684.15	783.54	742.17	1.32	-5.28	20.68	17	21
4	E021-06m2	70	430.88	430.88	0.00	716.45	726.02	804.37	783.99	1.34	-2.53	21.47	4	7
5	E022-04g2	100	375.28	375.28	0.00	423.60	446.74	502.36	498.76	5.46	-0.72	35.58	3	11
6	E022-06m2	70	495.85	495.85	0.00	898.26	906.72	980.92	958.73	0.94	-2.26	19.34	21	39
7	E023-03g2	150	715.02	715.02	0.00	942.30	985.61	1081.31	1067.04	4.60	-1.32	20.14	17	19
8	E023-05s2	150	665.17	665.17	0.00	1394.57	1412.34	1474.20	1441.51	1.27	-2.22	19.50	35	47
9	E026-08m2	60	607.65	607.65	0.00	1046.80	1048.56	1184.84	1167.27	0.17	-1.48	19.96	42	38
10	E030-03g2	150	667.42	667.42	0.00	729.24	763.79	803.10	793.16	4.74	-1.24	4.65	63	86
11	E030-04s2	110	664.48	666.15	0.25	1315.00	1318.35	1419.24	1361.81	0.25	-4.05	25.91	39	89
12	E031-09h2	100	610.00	610.00	0.00	614.33	618.04	774.70	727.68	0.60	-6.07	18.62	27	63
13	E033-03n2	640	2502.65	2504.52	0.07	537.87	538.39	549.44	546.95	0.10	-0.45	74.56	72	94
14	E033-04g2	190	1029.34	1034.98	0.55	1940.61	1945.35	2053.84	2037.71	0.24	-0.79	35.94	82	108
15	E033-05s2	190	1001.51	1006.67	0.52	2152.94	2165.73	2266.17	2190.76	0.59	-3.33	34.80	75	93
16	E036-11h2	70	698.61	703.17	0.65	976.56	979.36	1125.43	1022.79	0.29	-9.12	6.64	101	135
17	E041-14h2	70	861.79	867.85	0.70	1226.75	1236.72	1495.90	1484.21	0.81	-0.78	16.44	137	125
18	E045-04f2	115	987.10	987.10	0.00	1650.39	1673.82	1786.97	1747.15	1.42	-2.23	41.57	274	304
19	E051-05e2	80	723.93	726.96	0.42	939.36	943.57	1219.46	1175.09	0.45	-3.64	4.19	239	286
20	E072-04f2	50	488.69	494.83	1.26	518.10	521.46	630.63	625.35	0.65	-0.84	1.61	173	163
21	E076-07s2	100	964.49	982.97	1.92	1166.79	1172.06	1386.94	1358.49	0.45	-2.05	35.30	227	257
22	E076-08s2	80	976.70	996.91	2.07	1356.48	1395.46	1748.62	1691.18	2.87	-3.28	7.81	257	284
23	E076-10e2	80	984.00	986.79	0.28	1329.80	1353.46	1704.78	1662.96	1.78	-2.45	14.92	189	196
24	E076-14s2	80	1140.13	1166.13	2.28	1427.90	1438.41	1825.57	1769.29	0.74	-3.08	4.52	204	257
25	E101-08e2	100	1345.89	1373.14	2.02	1548.31	1583.37	1801.07	1794.86	2.26	-0.34	5.81	162	171
26	E101-10c2	90	1257.00	1269.59	1.00	1443.57	1467.53	1916.55	1874.53	1.66	-2.19	7.56	301	328
27	E101-14s2	90	1271.10	1304.47	2.63	1600.45	1713.43	1875.95	1842.83	7.06	-1.77	7.01	386	351
28	E121-07c2	120	2491.86	2526.50	1.39	3956.58	3993.91	4341.16	4257.76	0.94	-1.92	34.47	724	756
29	E135-07f2	150	2129.10	2177.96	2.29	2900.87	2937.53	3170.06	3110.19	1.26	-1.89	35.70	802	883
30	E151-12b2	70	1740.87	1776.86	2.07	2493.14	2695.54	3203.48	3070.42	8.12	-4.15	4.85	634	728
31	E200-16b2	60	2162.88	2201.82	1.80	3182.22	3295.28	3963.99	3872.45	3.55	-2.31	4.33	598	532
32	E200-17b2	70	2165.96	2203.06	1.71	3153.72	3193.56	3965.85	3917.71	1.26	-1.21	4.19	663	735
33	E200-17c2	70	2157.23	2203.10	2.13	3219.65	3247.90	3938.06	3896.32	0.88	-1.06	3.79	482	583
34	E241-22k2	30	1121.67	1154.81	2.95	1198.87	1206.14	1576.09	1571.11	0.61	-0.32	3.79	520	593
35	E253-27k2	30	1310.33	1364.02	4.10	1419.80	1483.58	1811.22	1799.64	4.49	-0.64	8.54	693	793
36	E256-14k2	35	1625.42	1685.63	3.70	3740.36	3794.69	4844.67	4815.52	1.45	-0.60	6.36	683	752
Average:					1.08					1.91	-2.39	17.68	249	276

Table 2: Results for Class 3 benchmark instances considering deterministic and stochastic travel times.

			Without Penalties			With Penalties								
#	Instance	τ	BKS [1]	OBS [2]	Δ (%) [2-1]	OBD_{det} [3]	OBS_{det} [4]	OBD_{stoch} [5]	OBS_{stoch} [6]	Δ (%) [4-3]	Δ (%) [6-5]	Δt_{OBS} [7]	CPU_{OBD} (s) [8]	CPU_{OBS} (s) [9]
1	E016-03m3	85	282.95	282.95	0.00	373.78	402.33	512.55	477.23	7.64	-6.89	19.08	4	7
2	E016-05m3	65	352.16	352.15	0.00	513.00	538.96	649.71	628.41	5.06	-3.28	16.81	42	48
3	E021-04m3	65	385.32	385.31	0.00	673.31	693.31	754.26	744.28	2.97	-1.32	24.32	2	25
4	E021-06m3	65	430.88	430.88	0.00	711.82	721.19	824.07	783.99	1.32	-4.86	22.42	30	35
5	E022-04g3	65	379.94	379.94	0.00	717.09	725.17	793.32	784.83	1.13	-1.07	38.03	3	6
6	E022-06m3	70	498.16	498.16	0.00	817.75	874.73	942.74	926.76	6.97	-1.69	19.53	0	1
7	E023-03g3	150	664.96	664.96	0.00	905.41	912.92	991.55	981.74	0.83	-0.99	25.58	32	40
8	E023-05s3	150	738.43	738.43	0.00	1019.86	1031.67	1087.79	1072.72	1.16	-1.39	25.70	23	28
9	E026-08m3	60	607.65	607.65	0.00	1046.80	1070.46	1185.08	1167.17	2.26	-1.51	20.02	2	3
10	E030-03g3	150	615.68	615.68	0.00	698.37	724.82	873.81	784.59	3.79	-10.21	3.74	47	53
11	E030-04s3	60	699.35	699.35	0.00	1029.28	1031.65	1163.04	1121.44	0.23	-3.58	24.66	10	16
12	E031-09h3	60	610.00	610.00	0.00	991.92	1017.38	1178.27	1129.16	2.57	-4.17	18.05	11	66
13	E033-03n3	385	2377.39	2377.39	0.00	3340.92	3340.92	3428.95	3418.21	0.00	-0.31	75.92	20	181
14	E033-04g3	180	988.79	989.22	0.04	1282.08	1282.46	1407.48	1382.42	0.03	-1.78	33.66	107	170
15	E033-05s3	190	1116.07	1120.75	0.42	1471.37	1474.19	1561.66	1559.08	0.19	-0.17	28.68	88	254
16	E036-11h3	190	698.61	698.61	0.00	897.36	907.45	1063.10	1024.35	1.12	-3.64	6.42	12	14
17	E041-14h3	80	861.79	861.79	0.00	1092.35	1174.67	1205.84	1103.94	7.54	-8.45	11.37	164	218
18	E045-04f3	80	986.30	986.30	0.00	1454.46	1466.81	1641.21	1586.44	0.85	-3.34	27.84	15	25
19	E051-05e3	85	749.43	752.33	0.39	888.90	912.34	1159.85	1093.22	2.64	-5.75	3.37	24	103
20	E072-04f3	60	521.31	521.31	0.00	514.43	521.03	795.50	780.40	1.28	-1.90	0.99	245	280
21	E076-07s3	85	1086.72	1095.68	0.82	2045.08	2062.23	2407.89	2362.38	0.84	-1.89	32.81	101	107
22	E076-08s3	85	1024.11	1031.78	0.75	1321.85	1359.79	1680.42	1575.10	2.87	-6.27	6.67	260	305
23	E076-10e3	75	1041.60	1059.41	1.71	1468.96	1527.53	1870.51	1759.12	3.99	-5.96	13.46	255	203
24	E076-14s3	85	1066.15	1070.53	0.41	1269.30	1303.14	1673.07	1551.30	2.67	-7.28	4.73	98	102
25	E101-08e3	100	1333.64	1352.19	1.39	1713.65	1731.83	2100.12	2015.88	1.06	-4.01	7.92	293	299
26	E101-10c3	110	1311.11	1337.12	1.98	1704.94	1759.47	2102.88	2072.10	3.20	-1.46	8.81	165	290
27	E101-14s3	85	1329.33	1360.98	2.38	1674.60	1686.83	2014.46	1965.96	0.73	-2.41	7.57	293	134
28	E121-07c3	110	2541.02	2594.43	2.10	4168.95	4168.96	4614.96	4563.87	0.00	-1.11	36.65	911	983
29	E135-07f3	85	2040.83	2058.47	0.86	3232.88	3232.88	3592.39	3581.42	0.00	-0.31	30.72	806	826
30	E151-12b3	85	1767.72	1809.25	2.35	2087.79	2125.99	2639.78	2532.84	1.83	-4.05	4.44	688	801
31	E200-16b3	85	2196.26	2237.84	1.89	2579.99	2595.15	3218.83	3106.86	0.59	-3.48	3.94	578	855
32	E200-17b3	85	2166.18	2223.25	2.63	2543.98	2650.49	3299.56	3135.29	4.19	-4.98	4.65	724	761
33	E200-17c3	85	2276.31	2323.04	2.05	2612.63	2621.28	3264.25	3221.29	0.33	-1.32	3.13	943	984
34	E241-22k3	20	1165.57	1187.22	1.86	2834.40	2956.95	4329.24	4249.26	4.32	-1.85	3.00	760	832
35	E253-27k3	40	1393.90	1443.85	3.58	1737.58	1792.47	3174.95	3042.49	3.16	-4.17	14.02	567	574
36	E256-14k3	45	1708.05	1772.05	3.75	2465.86	2493.98	3743.28	3681.75	1.14	-1.64	4.07	703	739
Average:					0.87					2.24	-3.29	17.58	251	288

Table 3: Results for Class 4 benchmark instances considering deterministic and stochastic travel times.

			Without Penalties			With Penalties								
#	Instance	τ	BKS [1]	OBS [2]	Δ (%) [2-1]	OBD_{det} [3]	OBS_{det} [4]	OBD_{stoch} [5]	OBS_{stoch} [6]	Δ (%) [4-3]	Δ (%) [6-5]	Δt_{OBS} [7]	CPU_{OBD} (s) [8]	CPU_{OBS} (s) [9]
1	E016-03m4	70	282.95	282.95	0.00	360.09	383.28	508.72	488.77	6.44	-3.92	1.38	0	0
2	E016-05m4	65	334.96	334.96	0.00	513.00	542.96	649.71	628.35	5.84	-3.29	9.59	5	6
3	E021-04m4	65	358.4	358.4	0.00	598.10	626.28	728.70	699.98	4.71	-3.94	20.89	15	16
4	E021-06m4	50	447.37	447.37	0.00	824.32	839.24	931.77	910.97	1.81	-2.23	32.18	2	3
5	E022-04g4	85	383.87	383.87	0.00	818.11	857.00	943.34	933.96	4.75	-0.99	20.10	2	25
6	E022-06m4	70	498.32	498.32	0.00	584.63	598.13	670.51	667.63	2.31	-0.43	23.16	42	62
7	E023-03g4	100	686.26	686.25	0.00	1196.96	1204.57	1313.86	1276.67	0.64	-2.83	58.92	2	4
8	E023-05s4	90	692.47	692.47	0.00	1227.98	1236.71	1351.77	1316.59	0.71	-2.60	67.31	66	82
9	E026-08m4	65	625.1	625.09	0.00	1075.40	1078.88	1214.64	1171.98	0.32	-3.51	22.96	25	39
10	E030-03g4	100	703.64	703.64	0.00	753.12	758.22	913.40	816.49	0.68	-10.61	2.23	7	69
11	E030-04s4	110	771.93	772.64	0.09	1104.57	1121.90	1236.69	1209.01	1.57	-2.24	20.58	24	38
12	E031-09h4	60	610.23	614.23	0.66	1012.63	1031.42	1209.54	1147.89	1.86	-5.10	18.28	11	146
13	E033-03n4	400	2533.79	2539.95	0.24	3291.58	3291.58	3379.85	3375.14	0.00	-0.14	67.41	56	83
14	E033-04g4	190	955.09	955.09	0.00	1232.80	1250.92	1340.75	1328.76	1.47	-0.89	28.29	58	69
15	E033-05s4	190	1164.63	1164.63	0.00	1492.54	1492.54	1584.72	1583.66	0.00	-0.07	26.93	51	90
16	E036-11h4	70	703.35	703.35	0.00	897.36	910.27	1063.30	1025.64	1.44	-3.54	6.95	22	31
17	E041-14h4	80	861.79	861.79	0.00	982.35	994.34	1097.59	1084.94	1.22	-1.15	12.43	47	82
18	E045-04f4	115	1100.52	1100.52	0.00	1607.46	1631.26	1859.96	1770.95	1.48	-4.79	33.54	238	216
19	E051-05e4	85	747.03	759.98	1.73	893.20	941.81	1136.19	1079.00	5.44	-5.03	4.61	96	254
20	E072-04f4	60	533.77	533.77	0.00	546.05	550.79	791.45	771.58	0.87	-2.51	0.94	164	171
21	E076-07s4	50	959.82	969.15	0.97	1859.41	1896.95	2238.07	2145.90	2.02	-4.12	32.25	126	144
22	E076-08s4	85	1041.8	1.060.00	1.75	1265.68	1284.08	1644.02	1549.87	1.45	-5.73	5.34	153	235
23	E076-10e4	75	1047.32	1.065.39	1.73	1472.09	1499.21	1820.41	1782.30	1.84	-2.09	11.51	295	299
24	E076-14s4	85	1086.09	1.098.00	1.10	1292.37	1310.57	1607.61	1570.22	1.41	-2.33	5.33	112	200
25	E101-08e4	85	1366.28	1.396.95	2.24	1713.65	1731.83	2100.12	2015.88	1.06	-4.01	6.43	293	299
26	E101-10c4	85	1362.22	1.388.74	1.95	1768.26	1770.48	2218.85	2189.37	0.13	-1.33	8.72	136	282
27	E101-14s4	85	1284.94	1.315.93	2.41	1648.01	1649.79	1940.89	1916.92	0.11	-1.24	7.71	107	231
28	E121-07c4	110	2580.29	2.623.32	1.67	4252.48	4260.46	4733.81	4728.33	0.19	-0.12	39.85	658	577
29	E135-07f4	85	2199.79	2.239.34	1.80	3507.83	3514.71	3869.95	3869.61	0.20	-0.01	30.06	795	939
30	E151-12b4	85	1784.14	1.829.99	2.57	2145.63	2159.63	2653.75	2510.54	0.65	-5.40	4.81	961	987
31	E200-16b4	70	2314.76	2.369.31	2.36	3165.54	3244.46	3898.71	3881.08	2.49	-0.45	8.30	531	730
32	E200-17b4	70	2206.72	2.258.73	2.36	3023.10	3194.84	3932.76	3837.61	5.68	-2.42	9.99	807	851
33	E200-17c4	40	2318.77	2.377.40	2.53	2706.17	2754.63	3419.42	3271.42	1.79	-4.33	2.96	686	785
34	E241-22k4	20	1163.96	1.183.45	1.67	2823.20	2921.13	4446.98	4312.15	3.47	-3.03	13.09	562	812
35	E253-27k4	40	1452.59	1.493.56	2.82	1795.64	1871.11	3285.73	3199.61	4.20	-2.62	3.58	784	847
36	E256-14k4	45	1605	1.645.43	2.52	2316.52	2330.78	3484.78	3454.95	0.62	-0.86	5.95	811	826
Average:					0.98					1.97	-2.77	18.74	243	293

Table 4: Results for Class 5 benchmark instances considering deterministic and stochastic travel times.

			Without Penalties			With Penalties								
#	Instance	τ	BKS [1]	OBS [2]	Δ (%) [2-1]	OBD_{det} [3]	OBS_{det} [4]	OBD_{stoch} [5]	OBS_{stoch} [6]	Δ (%) [4-3]	Δ (%) [6-5]	Δt_{OBS} [7]	CPU_{OBD} (s) [8]	CPU_{OBS} (s) [9]
1	E016-03m5	70	278.73	278.73	0.00	452.98	481.26	552.94	533.91	6.24	-3.44	0.99	0	0
2	E016-05m5	65	334.96	334.96	0.00	578.96	581.35	628.28	584.21	0.41	-7.01	16.74	9	11
3	E021-04m5	70	358.40	358.40	0.00	598.1	635.64	728.45	701.91	6.28	-3.64	35.63	18	21
4	E021-06m5	70	430.88	430.88	0.00	716.45	721.19	804.82	784.04	0.66	-2.58	21.47	13	12
5	E022-04g5	100	375.28	375.28	0.00	684.06	698.14	800.53	774.19	2.06	-3.29	35.58	9	14
6	E022-06m5	70	495.85	495.85	0.00	817.75	850.56	942.73	927.51	4.01	-1.61	22.82	36	76
7	E023-03g5	150	657.77	657.77	0.00	1112.63	1148.96	1227.75	1192.82	3.27	-2.85	23.81	48	67
8	E023-05s5	150	609.90	609.90	0.00	1060.06	1079	1142.72	1130.63	1.79	-1.06	37.16	42	49
9	E026-08m5	60	607.65	607.65	0.00	1112.63	1134.21	1227.75	1192.82	1.94	-2.85	19.74	31	73
10	E030-03g5	150	678.62	680.37	0.26	825.42	850.05	948.91	893.81	2.98	-5.81	1.15	14	38
11	E030-04s5	110	624.82	624.82	0.00	924.9	954.14	1062.56	1025.87	3.16	-3.45	29.92	25	47
12	E031-09h5	100	610.00	610.00	0.00	615.6	616.45	667.42	661.55	0.14	-0.88	18.02	67	82
13	E033-03n5	640	2334.78	2338.92	0.18	2545.46	2549.51	2608.29	2576.07	0.16	-1.24	27.53	59	83
14	E033-04g5	190	871.22	894.96	2.72	1196.45	1209.46	1279.47	1273.42	1.09	-0.47	33.56	84	102
15	E033-05s5	190	1159.94	1161.70	0.15	1529.18	1534.55	1660.41	1635.28	0.35	-1.51	22.70	76	94
16	E036-11h5	70	698.61	698.61	0.00	897.35	907.45	1064.43	1023.79	1.13	-3.82	6.55	46	91
17	E041-14h5	70	861.79	863.26	0.17	956.14	959.43	1111.86	1109.99	0.34	-0.17	14.64	72	85
18	E045-04f5	115	917.94	925.49	0.82	1468.15	1468.15	1590.93	1588.95	0.00	-0.12	46.22	168	186
19	E051-05x5	80	644.59	655.06	1.62	936.71	957.67	1179.02	1122.13	2.24	-4.83	10.77	184	243
20	E072-04f5	80	466.79	480.46	2.93	489.22	516.82	600.38	591.08	5.64	-1.55	1.69	172	182
21	E076-07s5	100	870.82	903.64	3.77	1286.49	1295.67	1589.41	1536.6	0.71	-3.32	36.55	162	158
22	E076-08s5	80	928.02	943.99	1.72	1244.79	1261.05	1579.9	1462.1	1.31	-7.46	9.76	204	308
23	E076-10e5	80	922.34	953.82	3.41	1137.06	1139.09	1439.43	1401.8	0.18	-2.61	17.16	305	347
24	E076-14s5	80	1042.37	1053.71	1.09	1170.83	1173.48	1535.88	1443.18	0.23	-6.04	6.39	155	237
25	E101-08e5	100	1149.66	1191.01	3.60	1363.54	1365.67	1778.56	1637.35	0.16	-7.94	9.86	262	302
26	E101-10c5	90	1209.34	1238.67	2.43	1554.28	1562.27	2004.43	1892.63	0.51	-5.58	9.82	401	473
27	E101-14s5	90	1231.52	1271.12	3.22	1559.68	1565.64	1986.09	1828.96	0.38	-7.91	8.85	394	381
28	E121-07c5	120	2276.71	2352.40	3.32	3843.08	3875.83	4277.52	4230.02	0.85	-1.11	41.19	483	526
29	E135-07f5	150	2115.53	2167.53	2.46	3088.41	3093.35	3417.82	3275.44	0.16	-4.17	36.17	736	861
30	E151-12b5	70	1512.71	1571.10	3.86	2416	2441.37	2991.84	2958.42	1.05	-1.12	6.45	835	825
31	E200-16b5	60	1968.89	2040.45	3.63	3356.79	3396.45	4200.91	4072	1.18	-3.07	4.53	534	682
32	E200-17b5	70	1938.96	2013.38	3.84	3172.81	3288.62	4032.65	3876.16	3.65	-3.88	5.77	703	794
33	E200-17c5	70	1946.51	2038.37	4.72	3341.76	3361.37	4150.91	4080.16	0.59	-1.70	5.03	703	812
34	E241-22k5	30	1006.38	1061.72	5.50	2018.68	2048.36	2455.31	2371.15	1.47	-3.43	16.09	648	809
35	E253-27k5	30	1224.21	1284.32	4.91	1649.45	1694.81	2053.3	1998.91	2.75	-2.65	6.02	793	857
36	E256-14k5	35	1457.05	1536.49	5.45	1738.45	1756.43	1948.03	1925.25	1.03	-1.17	7.59	834	911
Average:					1.83					1.67	-3.20	18.16	259	301

in the best-known solution for the deterministic version. For each instance, the associated threshold value (τ) is also reported in each table. As described in Section 3, companies often need to deal with unexpected costs due to longer-than-expected working hours. This characteristic was captured by modifying the standard objective function into a non-smooth expression that penalises excess time above a given threshold, as given in Equations (12) and (11). For our experiments, and in order to obtain numerical results but without losing generality, the overtime penalty parameter ρ was set to 1.25, while the parameter γ was set to 40.

First, we validate our approach on the original 2L-VRP instances, that is, without considering stochastic travel times or overtime penalties. We report in Tables 1 to 4 our obtained best solution (*OBS [2]*) for each deterministic instance, as well as the best known solution in the literature (*BKS [1]*) and the corresponding gap (Δ*(%) [2-1]*). In particular, we measure the performance of our algorithm using the current best solutions reported by Wei et al. (2018b). We observe that our method is able to match or remain very close to the best-known solutions for each deterministic instance, with average gaps around 1%. Despite aimed at solving stochastic instances, results show that our algorithm is comparable to other state-of-the-art approaches for the deterministic 2L-VRP.

The second part of the tables report the results obtained for the 2L-SVRP stochastic instances considering random travel times and overtime penalties. These figures correspond to the best found solution over 5 independent executions of our algorithm per instance, allowing a maximum running time of 1,000 seconds per run. In all cases, the average costs in the deterministic scenario for the best deterministic (OBD_{det} *[3]*) and stochastic (OBS_{det} *[4]*) solutions are presented. Likewise, we include the average values in the stochastic case for the best deterministic (OBD_{stoch} *[5]*) and stochastic (OBS_{stoch} *[6]*) solutions. The average excess time incurred by the best stochastic solution in the stochastic scenario is also provided in the table (Δt_{OBS} *[7]*). Finally, the two last columns present the computational time required to reach the deterministic (CPU_{OBD} *[8]*) and stochastic (CPU_{OBS} *[8]*) solutions.

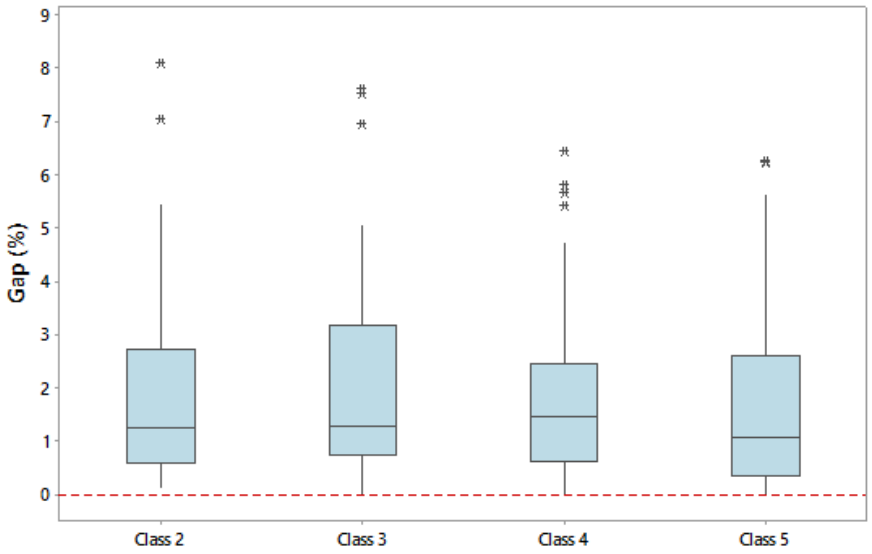
Our results show that the solutions provided by our approach for the 2L-SVRP clearly outperform the best-known solutions for the deterministic 2L-VRP when these are considered as solutions of the 2L-SVRP. In other words, near-optimal solutions for the deterministic version of the problem might be sub-optimal solutions for the stochastic version. Hence, the importance of

integrating simulation methods when dealing with stochastic optimization problems. As expected, considering the stochastic nature of some variables during the searching process yields worse deterministic solutions – an average gap of 1.94%, obtaining a maximum average gap of 2.24% for class 3, when compared to the deterministic approach. However, these same solutions show a better performance when variability is considered, with an average gap of -2.91%, obtaining a maximum average gap of -3.29% for class 3, when compared to the best deterministic solutions applied in stochastic instances. Furthermore, this behaviour is consistent across instances of different classes, as shown in Figure 2. For some instances, we observe that using our approach may yield savings of up to 10% when these random variable travel times are considered (Figure 2b).

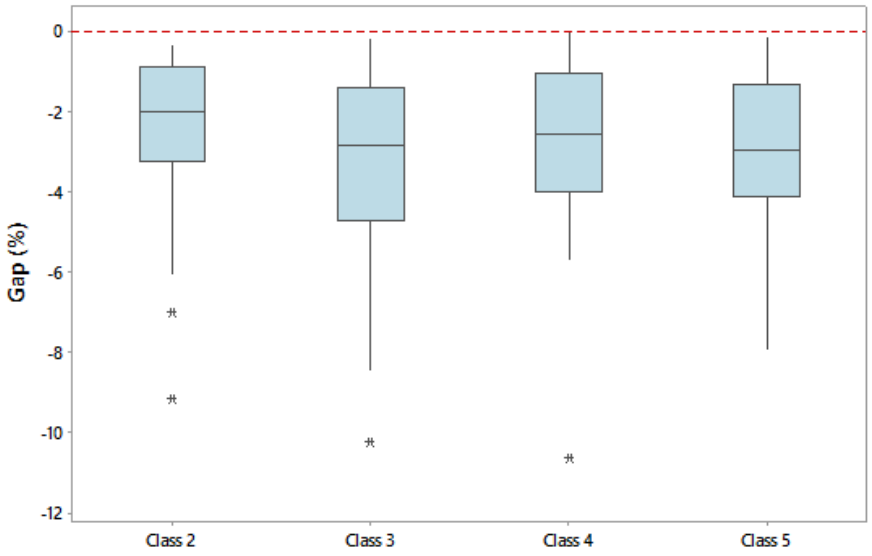
All in all, the results confirm that not accounting for variability during the search process may have a significant impact on the quality of the final solution. For instance, the direct application of deterministic solutions might provide a slightly lower cost in ideal situations without uncertainty (Figure 2a), but will cause a much higher cost in a scenario under uncertainty (Figure 2b). This is due to the penalty cost considered for overtime routes, which is a reasonable assumption in most real-life situations.

6. Conclusions

In this paper, we present a simheuristic algorithm for solving a stochastic variant of the two-dimensional vehicle routing problem (2L-VRP). This problem can be found in many real-life applications, like the one that motivated this work. In many cases, companies have to deal with uncertainty in some aspects of the problem, such as stochastic travel or service times, which typically derive in overtime penalty costs. However, this aspect has seldom been addressed in the 2L-VRP literature yet. Our simheuristic approach combines an iterated local search framework with biased-randomised versions of classical routing and packing heuristics. Both heuristics are integrated in the routing-construction method, ensuring packing feasibility for all routes during the construction process, contrary to more traditional two-stage approaches. In order to deal with the stochastic nature of the proposed problem, Monte Carlo simulation is integrated at two stages of the metaheuristic framework. This allows us to account for uncertainty in travel times during the search. The algorithm is enhanced by fast-access memory-based techniques, which help to reduce computational times. To test our



(a) Deterministic scenarios



(b) Stochastic scenarios

Figure 2: Gap (%) distribution between the best stochastic and best deterministic solution, in equivalent *(a)* deterministic and *(b)* stochastic instances of the 2L-VRP.

methodology, stochastic instances of the 2L-VRP have been generated by modifying the ones contained in well-known deterministic benchmark sets. Random delays in base travel times are modelled using Log-Normal probability distributions. The numerical results show that our methodology is capable of generating much better solutions for scenarios under uncertainty, while still being competitive with state-of-the-art approaches for the classical deterministic scenarios without uncertainty. Moreover, our approach is able to perform consistently across all tested instances. As future research, we plan to extend our methodology to tackle other stochastic 2L-VRP variants (e.g., with sequential loading or backhauls). Indeed, the use of other metaheuristic frameworks and their integration with simulation techniques seems quite promising to us. Therefore, it constitutes an interesting research line to pursue for stochastic 2L-VRP variants as well as similar stochastic optimization problems, where more traditional methods such as stochastic programming might be of limited employability due to the complex nature of the underlying optimization problem and the large-size of realistic instances.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (TRA2015-71883-REDT), FEDER. and the Erasmus+ programme (2018-1-ES01-KA103-04976).

References

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

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., and Juan, A. A. (2014). Rich vehicle routing problem: Survey. *ACM Computing Surveys*, 47(2):1–28.

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

de Armas, J., Juan, A. A., Marquès, J. M., and Pedroso, J. P. (2016). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, pages 1–16.

Dominguez, O., Guimarans, D., Juan, A. A., and Nuez, I. (2016a). A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, 255(2):442–462.

Dominguez, O., Juan, A. A., Barrios, B., Faulin, J., and Agustin, A. (2014a). Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, doi: 10.1007/s10479-014-1551-4.

Dominguez, O., Juan, A. A., and Faulin, J. (2014b). A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. *International Transactions in Operational Research*, 21:375–398.

Dominguez, O., Juan, A. A., Nuez, I., and Ouelhadj, D. (2016b). An ILS-biased randomization algorithm for the two-dimensional loading HFVRP with sequential loading and items rotations. *Journal of the Operational Research Society*, 67(1):37–53.

Duhamel, C., Lacomme, P., Quilliot, A., and Toussaint, H. (2009). 2L-VRP: a GRASP resolution scheme based on RCPSP. In *Proceedings of International Conference on Computers and Industrial Engineering*, pages 1094–1099, Troyes, France.

Duhamel, C., Lacomme, P., Quilliot, A., and Toussaint, H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38(3):617–640.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., and Rousseau, L.-M. (2016a). A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *European Journal of Operational Research*, 249:55–66.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., and Rousseau, L.-M. (2016b). The vehicle routing problem with hard time windows and stochastic service times. *EURO Journal on Transportation and Logistics*, pages 1–29.

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

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*, 51:4–18.

Golden, B., Raghavan, S., and Wasil, E., editors (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, NY.

Gonzalez-Martin, S., Juan, A. A., Riera, D., Elizondo, M. G., and Ramos, J. J. (2018). A simheuristic algorithm for solving the arc routing problem with stochastic demands. *Journal of Simulation*, 12(1):53–66.

Gonzalez-Neira, E. M., Ferone, D., Hatami, S., and Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 79:23–36.

Grasas, A., Juan, A. A., Faulin, J., de Armas, J., and Ramalhinho, H. (2017). Biased randomization of heuristics using skewed probability distributions: a survey and some applications. *Computers & Industrial Engineering*, 110:216–228.

Grasas, A., Juan, A. A., and Lourenço, H. R. (2016a). Simils: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1):69–77.

Grasas, A., Juan, A. A., and Lourenço, H. R. (2016b). Simils: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1):69–77.

Gruler, A., Fikar, C., Juan, A. A., Hirsch, P., and Contreras-Bolton, C. (2017a). Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization. *Journal of Simulation*, 11(1):11–19.

Gruler, A., Panadero, J., de Armas, J., Moreno Pérez, J., and Juan, A. (2018a). Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. *Computers and Industrial Engineering*, 123:278–288.

Gruler, A., Panadero, J., de Armas, J., Pérez, J., and Juan, A. (2018b). A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands. *International Transactions in Operational Research*, doi 10.1111/itor.12540.

Gruler, A., Quintero-Araujo, C. amd Calvet, L., and Juan, A. A. (2017b). Waste collection under uncertainty: a simheuristic based on variable neighborhood search. *European Journal of Industrial Engineering*, 11(2):228–255.

Guimarans, D., Arias, P., and Mujica Mota, M. (2015). Large neighbourhood search and simulation for disruption management in the airline industry. In Mujica Mota, M., Flores, I., and Guimarans, D., editors, *Applied Simulation and Optimization: In Logistics, Industrial and Aeronautical Practice*, pages 169–201. Springer.

Guimarans, D., Dominguez, O., Juan, A. A., and Martinez, E. (2016). A multi-start simheuristic for the stochastic two-dimensional vehicle routing problem. In Roeder, T. M. K., Frazier, P. I., Szechtman, R., Zhou, E., Huschka, T., and Chick, E., editors, *Proceedings of the 2016 Winter Simulation Conference*, pages 2326–2334, Arlington, VA, USA.

Hatami, S., Calvet, L., Fernández-Viagas, V., Framiñán, J. M., and Juan, A. A. (2018). A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem. *Simulation Modelling Practice and Theory*, 86:55–71.

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

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

Juan, A. A., Barrios, B., Vallada, E., Riera, D., and Jorba, J. (2014a). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 46:101–117.

Juan, A. A., Faulin, J., Ferrer, A., Lourenço, H. R., and Barrios, B. (2013a). Mirha: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *TOP*, 21(1):109–132.

Juan, A. A., Faulin, J., Grasman, S., Rabe, M., and Figueira, G. (2015). A review of simheuristics: extending metaheuristics to deal with stochastic optimization problems. *Operations Research Perspectives*, 2:62–72.

Juan, A. A., Faulin, J., Grasman, S., Riera, D., Marull, J., and Mendez, C. (2011a). Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. *Transportation Research Part C*, 19(5):751–765.

Juan, A. A., Faulin, J., Jorba, J., Caceres, J., and Marquès, J. M. (2013b). Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands. *Annals of Operations Research*, 207(1):43–65.

Juan, A. A., Faulin, J., Jorba, J., Riera, D., Masip, D., and Barrios, B. (2011b). On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics. *Journal of the Operational Research Society*, 62(6):1085–1097.

Juan, A. A., Faulin, J., Ruiz, R., Barrios, B., and Caballé, S. (2010). The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. *Applied Soft Computing*, 10(1):215–224.

Juan, A. A., Grasman, S., Caceres-Cruz, J., and Bektas, T. (2014b). A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. *Simulation Modelling Practice and Theory*, 46:40–52.

Kenyon, A. S. and Morton, D. P. (2003). Stochastic vehicle routing with random travel times. *Transportation Science*, 37(1):69–82.

Lahyani, R., Khemakhem, M., and Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1):1–14.

Lambert, V., Laporte, G., and Louveaux, F. (1993). Designing collection routes through bank branches. *Computers & Operations Research*, 20(7):783–791.

Laporte, G., Louveaux, F., and Mercure, H. (1992). The vehicle routing problem with stochastic travel times. *Transportation Science*, 26(3):161–170.

Lecluyse, C., Van Woensel, T., and Peremans, H. (2009). Vehicle routing with stochastic time-dependent travel times. *4OR: A Quarterly Journal of Operations Research*, 7:363–377.

Leung, S., 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., 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.

Lourenço, H. R., Martin, O. C., and Stützle (2010). Iterated Local Search: Framework and appllications. In Gendreau, M. and Potvin, J., editors, *Handbook of Metaheuristics*, pages 363–397. Springer, New York, NY.

Panadero, J., Doering, J., Kizys, R., Juan, A. A., and Fito, A. (2018). A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics*, pages 1–23.

Rostami, B., Desaulniers, G., Errico, F., and Lodi, A. (2017). The vehicle routing problem with stochastic and correlated travel times. Technical report, Polytechnique Montréal.

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

Wang, F., Tao, Y., and Shi, N. (2009). A survey on vehicle routing problem with loading constraints. In *Proceedings of International Joint Conference on Computational Sciences and Optimization*, pages 602–606, Sanya, China.

Wei, L., Zhang, Z., Zhang, D., and Leung, S. (2018a). 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., and Leung, S. C. (2018b). 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., Tarantilis, C., and Kiranoudis, C. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195:729–743.

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

Zachariadis, E., Tarantilis, C., and Kiranoudis, C. (2016). The vehicle routing problem with simultaneous pick-ups and deliveries and two-dimensional loading constraints. *European Journal of Operational Research*, 251:369–386.

Zachariadis, E., Tarantilis, C., and Kiranoudis, C. (2017). Vehicle routing strategies for pick-up and delivery service under two dimensional loading constraints. *Operational Research*, 17:115–143.

Zhang, J., Lam, W. H. K., and Chen, B. Y. (2013). A stochastic vehicle routing problem with travel time uncertainty: Trade-off between cost and customer service. *Networks and Spatial Economics*, 13:471–496.