A Simheuristic Algorithm for the Capacitated Location Routing Problem with Stochastic Demands

Carlos L. Quintero-Araujo[a,*], Daniel Guimarans[b], Angel A. Juan[c]

[a] *Operations & Supply Chain Management Research Group – International School of Economics and Administrative Sciences – Universidad de La Sabana Chia - Colombia*
[b] *Faculty of Information Technology – Monash University, Melbourne - Australia*
[c] *IN3 – Computer Science Dept. – Universitat Oberta de Catalunya, Barcelona - Spain*

Abstract

The capacitated location routing problem (CLRP) integrates a facility location problem with a multi-depot vehicle routing problem. In this paper, we consider the CLRP with stochastic demands, whose specific values are only revealed once a vehicle visits each customer. The main goal is then to minimize the expected total cost, which includes not only the costs of opening facilities, using a fleet of vehicles, and executing a routing plan, but also the cost of applying corrective actions. These actions are required whenever a route failure occurs due to unexpectedly high demands in a route. To solve this stochastic and *NP-hard* optimization problem, a simheuristic algorithm is proposed. It hybridizes simulation with an iterated local search metaheuristic in order to: *(i)* propose a safety-stock policy to diminish the risk of suffering route failures; and *(ii)* estimate both the expected cost as well as the reliability index of each 'elite' solution found. The competitiveness of our approach is shown in a series of computational experiments, which make use of classical CLRP benchmarks. These benchmarks are also extended to consider scenarios under uncertainty. Different variability levels for the random demands are analyzed. Moreover, the effect of the safety-stock policy on the solution cost and reliability index is also discussed.

*Corresponding Author
Email addresses: carlosqa@unisabana.edu.co (Carlos L. Quintero-Araujo), daniel.guimarans@monash.edu (Daniel Guimarans), ajuanp@uoc.edu (Angel A. Juan)

Keywords: location routing problem, stochastic demands, simheuristics, iterated local search, biased randomization

1. Introduction

In logistics management, facility location and route planning are linked decisions. However, in most real-life situations, these decisions are made in a sequential and independent way. On the one hand, size (capacity) and location of warehousing facilities or depots are usually decided without considering its effects on routing plans. On the other hand, customers are usually allocated to facilities just considering dedicated trips (i.e., minimizing allocation distance), but without considering its effects for routing purposes. Therefore, such isolated decisions lead to suboptimal solutions. To overcome this situation, several authors address the integrated capacitated location routing problem, or CLRP (Prodhon and Prins, 2014; Quintero-Araujo et al., 2017a).

As illustrated in Figure 1, the aim of the CLRP is to determine: *(i)* the facilities to be opened among a set of potential candidates with different locations and capacities; *(ii)* the allocation of customers to the open facilities; and, *(iii)* the corresponding routes that serve all customers' demands. Usually, the main goal is to minimize the total cost, i.e., the addition of opening costs, fleet costs, and routing costs. In practical terms, the CLRP combines the facility location problem (FLP) with the multi-depot vehicle routing problem (MDVRP). Since both of these problems are *NP-hard*, it is clear that their integration in the CLRP is also an *NP-hard* optimization problem.

While uncertainty is one of the main characteristics of real-life problems, most of the published works analyze the deterministic version of the CLRP, in which all inputs and parameters are supposed to be known in advance. To contribute to close this gap, this paper deals with a more realistic version of the CLRP with stochastic demands (CLRPSD). The introduction of stochastic demands, whose specific value is only revealed once a vehicle reaches each customer on its route, might also result in route failures. In effect, whenever the accumulated demand requested by the customers on a route exceeds the vehicle capacity, the route cannot be completed as designed and a costly corrective action will be required, e.g., a round-trip to the depot facility for a truck reload. As illustrated in the recent work by Zhang et al. (2019), the

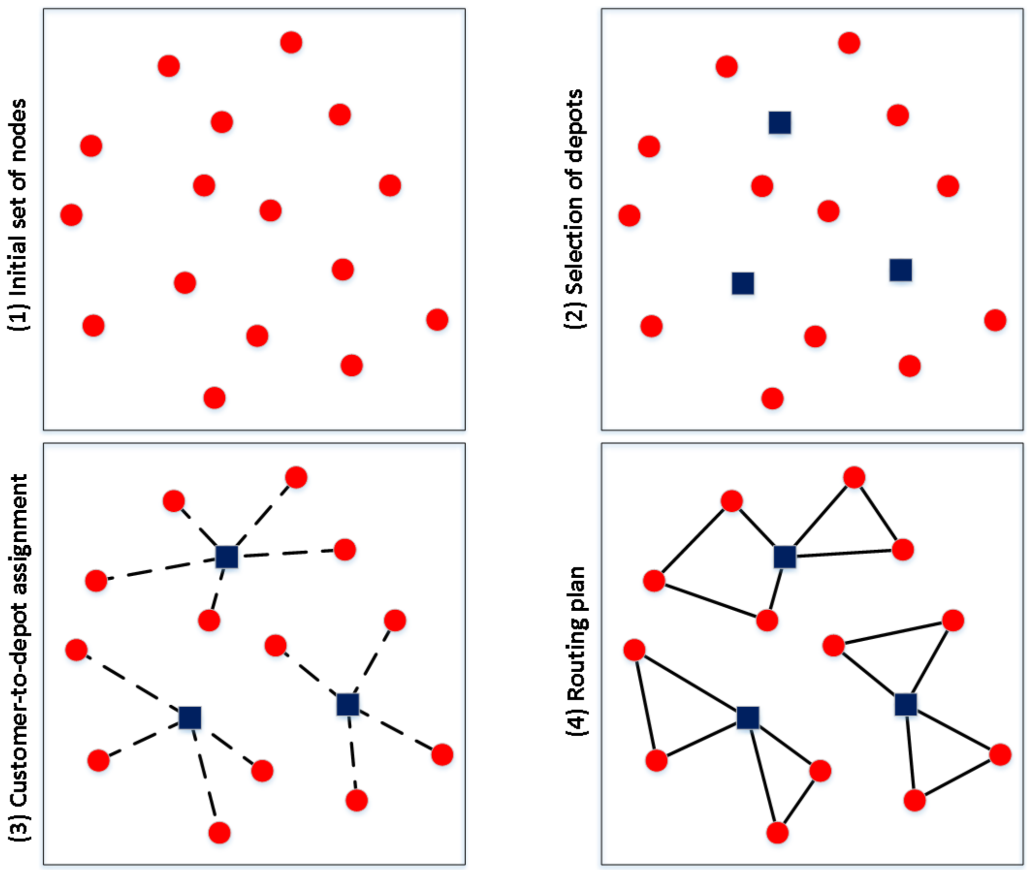

Figure 1: An illustrative description of the CLRP.

study of the CRLPSD is relevant, among other potential applications, when dealing with fleets of electric vehicles in the context of smart cities.

In order to minimize the expected total cost in the CLRPSD, we propose a simheuristic algorithm combining Monte Carlo simulation (MCS) with a metaheuristic framework. In our case, simulation is integrated into an iterated local search framework (SimILS), as proposed in Grasas et al. (2016). The resulting SimILS also makes use of different perturbation operators and biased randomization techniques. As discussed in Grasas et al. (2017), biased randomization techniques allow to introduce a 'skewed' (non-uniform) random behavior into heuristic-based procedures, thus orienting the search process towards promising regions of the solution space (Fikar et al., 2016). Finally, safety stocks in each vehicle are considered during the routing design stage in order to diminish the risk of suffering route failures during the execution of the routing plan. Accordingly, the main contributions of this work are: *(i)* a formal description of the CLRPSD, which extends the deterministic CLRP by also considering non-smooth costs generated by the application of corrective actions; *(ii)* an original simheuristic algorithm, combining simulation with a biased-randomized iterated local search to solve the CLRPSD; *(iii)* the incorporation of reliability indexes (to measure the probability that a given solution does not suffer route failures) and safety-stock levels during the routing design stage; *(iv)* the analysis of how solutions change as the level of uncertainty is increased; and, *(v)* a reliability / risk analysis on alternative solutions, so that different decision-maker profiles can be fitted.

The remainder of this paper is organized as follows. Section 2 contains a literature review on related work. Section 3 describes the problem under study. Sections 4 and 5 present, respectively, the solving approach used to tackle the CLRPSD, as well as the numerical experiments carried out. Finally, section 6 outlines some conclusions and further research opportunities.

2. Literature Review

Since the number of works related to the CLRPSD is still quite scarce, this section reviews first the related work on the deterministic CLRP. Then, different works on stochastic versions of the problem are reviewed, and a short discussion on the concept of simheuristics is provided.

2.1. Related Works on the Deterministic CLRP

The CLRP comprises all decision levels (strategic, tactical, and operational) in supply chain management. Strategic decisions are related to the number and size of facilities to be opened, while tactical and operational ones are associated with customers' allocation to open facilities and the corresponding distribution routes, respectively. In terms of classical optimization problems, the CLRP combines the FLP —which is associated with strategic decisions— and the MDVRP —which is related to customers' allocation to facilities and the consequent route planning (Nagy and Salhi, 2007). The benefits derived from taking into account routing decisions while locating facilities were firstly estimated in Salhi and Rand (1989). The authors showed that solving the associated FLP and VRP sub-problems independently usually generates sub-optimal solutions. Due to computational constraints, the first works addressed the CLRP by firstly tackling the FLP and then using its solution to solve the associated MDVRP. However, recent approaches propose to solve the problem using an integrated perspective (Dai et al., 2019; Quintero-Araujo et al., 2017a). Lopes et al. (2016) proposed a hybrid genetic algorithm to solve the CLRP using two different local search strategies to improve both the location and the routing level. The CLRP has several real-life applications that include, among others, city logistics (Nataraj et al., 2019), horizontal cooperation (Quintero-Araujo et al., 2019), and location of battery swap stations for electric vehicle routing (Hof et al., 2017). Despite the importance of the CLRP in supply chain management, the number of published works is significantly lower than the number of articles related to other vehicle routing variants.

Due to the *NP-hard* nature of the two sub-problems that constitute the CLRP, exact methods are less frequent than heuristic-based approaches. Among the former, Belenguer et al. (2011) and Akca et al. (2009) solved instances with up to 50 customers and 5-10 facilities, whilst Baldacci et al. (2011) and Contardo et al. (2014) were capable of solving instances with at most 200 customers and 10-14 potential facility locations. Constructive clustering-based heuristics to solve the CLRP have been proposed by Barreto et al. (2007), Boudahri et al. (2013), and Lopes et al. (2008). Concerning metaheuristic approaches, some population-based algorithms have been proposed (Prins et al., 2006a; Ting and Chen, 2013), although most published works make use of single-solution approaches (Prins et al., 2006b, 2007; Duhamel et al., 2010; Escobar et al., 2014; Contardo et al., 2014). Also, a number of CLRP applications have been studied in the literature.

For instance, Mousavi and Tavakkoli-Moghaddam (2013) analyzed a real-life CLRP related to cross-docking platforms.

The two subproblems composing the CLRP have received far more attention separately. For example, the FLP has been addressed using exact (Efroymson and Ray, 1966; Schrage, 1978) and heuristic methods (Hochbaum, 1982; Li, 2011; De Armas et al., 2016). Application examples range from reallocation of ambulances in real-time (Gendreau et al., 2001) to digital network design problems (Thouin and Coates, 2008). The reader is referred to Snyder and Daskin (2006) and Fotakis (2011) for a more detailed overview of solving methods and applications related to the FLP. As for the MDVRP, different heuristic approaches have been proposed by Tillman and Cain (1972), Gillett and Johnson (1976), and Golden et al. (1977). On the side of metaheuristics, different methods have been tested. These include tabu search (Cordeau et al., 1997; Renaud et al., 1996), genetic algorithms (Thangiah and Salhi, 2001; Ho et al., 2008), adaptive large neighborhood search (Pisinger and Ropke, 2007), and iterated local search (Juan et al., 2015), among others. The reader is referred to Montoya-Torres et al. (2015) for a recent review on the MDVRP.

2.2. Related Works on the Stochastic CLRP and Simheuristics

Regarding the CLRP with stochastic components, the number of existing works is very limited. Still, different uncertainty sources have been considered in the literature: travel times, customers' service request, customers' demands, etc. A variant of the CLRP with stochastic customer requests is analyzed in Albareda-Sambola et al. (2007). In this work, uncapacitated vehicles are used to perform routing tasks. Customers' request for service is not known in advance and it is modeled by means of a Bernoulli distribution. Probabilistic travel times are included in Ghaffari-Nasab et al. (2013). These authors solved a bi-objective CLRP in which the analyzed objectives are the total costs and the maximum delivery time to the customers. Stochastic demands have been modeled both by means of fuzzy numbers (Mehrjerdi and Nadizadeh, 2013; Zarandi et al., 2013) as well as random variables (Marinakis, 2015; Marinakis et al., 2016). Unfortunately, these works limit their analysis to the case in which the random demands follow a Poisson probability distribution, which not only is a discrete distribution but also has a fixed variance once the mean is given. Therefore, they do not consider scenarios with different uncertainty levels. Moreover, no reference to simulation techniques is made in these papers and, accordingly, they are methodologically different from the simheuristic algorithm presented here. More recently, Zhang et al. (2019) addressed a battery swap station LRP with stochastic demands using a modified version of Particle Swarm Optimization (PSO) combined with Variable Neighborhood Search (VNS). The former algorithm is used for the location level, while the latter is used for the route-design phase.

Different examples on the combination of simulation with heuristic-based methods for solving combinatorial optimization problems can be found in the literature (Faulin et al., 2008; Faulin and Juan, 2008). Simheuristics is a relatively new and efficient approach to tackle combinatorial optimization problems under uncertainty (Juan et al., 2018). Roughly speaking, a simheuristic algorithm works in the following way: *(i)* given a stochastic problem setting, the random variables are transformed into deterministic values by considering expected values; *(ii)* a metaheuristic framework is used to generate high-quality solutions for the deterministic instance that can also be 'promising' solutions for the stochastic version of the problem; *(iii)* these promising solutions are sent to a simulation component in order to estimate its quality in a stochastic environment —the simulation component also provides useful feedback to better guide the metaheuristic search; and, *(iv)* a refinement of the estimates is obtained for a subset of 'elite' solutions using a more computationally-intensive simulation process. Typically, a risk or reliability analysis of these elite solutions is also performed to include the risk-orientation profile of the decision maker. Different simheuristic algorithms have been presented in the literature to solve routing problems. Stochastic demands in vehicle routing problems are addressed in Juan et al. (2013) and Quintero-Araujo et al. (2017b). Similarly, the arc routing problem with stochastic demands is discussed in Gonzalez-Martin et al. (2018), while the waste collection problem with stochastic demands is analyzed in Gruler et al. (2017). Stochastic versions of the inventory routing problem can be found in Juan et al. (2014b), Gruler et al. (2018a), and Gruler et al. (2018b). Likewise, the stochastic two-dimensional vehicle routing problem is analyzed in Guimarans et al. (2018). In addition, simheuristics can also be used in the context of stochastic scheduling problems, like the ones proposed in Fu et al. (2018), Fu et al. (2019), Juan et al. (2014a), Gonzalez-Neira et al. (2017), and González-Neira et al. (2019).

3. Problem Description

In this section we present a model for the stochastic CLRPSD, which is an extension of the mixed integer linear programming model proposed by Prins et al. (2006b) for the deterministic CLRP. Consider a graph $G = (V, A, C)$, where:

- V is the set of all nodes, including: *(i)* a subset $\emptyset \neq I \subset V$ of customers with independent random demands $D_i > 0$ ($\forall i \in I$), each following a non-negative and upper-bounded probability distribution, but whose specific values are only revealed once the customer is visited; and, *(ii)* a subset $\emptyset \neq W \subset V$ of potential depot facilities with opening costs $o_w \geq 0$ and service capacities $q_w > 0$ ($\forall w \in W$).
- A is the set of oriented arcs a_{ij} linking each pair of distinct nodes $i, j \in V$.
- C is the matrix including the distance-based costs associated with traversing each arc, $c_{ij} = c_{ji} > 0$. It is assumed that the triangle inequality is satisfied.

Also, a virtually unlimited set K of homogeneous vehicles is available, each of them with a loading capacity $h > max\{D_i\}$ and a fixed utilization cost $u \geq 0$. We assume a percentage of this capacity can be reserved to store a safety stock ($\%SS$) to respond to demand variability. In this context, the following binary decision variables are considered:

- y_w, which takes the value 1 if facility $w \in W$ is open and 0 otherwise,
- x_{iw}, which takes the value 1 if customer $i \in I$ is assigned to facility $w \in W$ while it takes the value 0 otherwise, and
- f_{ak}, which takes the value 1 if arc $a \in A$ belongs to the route covered by vehicle $k \in K$, and 0 otherwise.

Notice that the total cost of a CLRPSD solution will be a random variable resulting from the aggregation of: *(i)* the cost of opening the selected facilities; *(ii)* the cost associated with the use of a given number of vehicles; and, *(iii)* the distance-based routing cost, which will depend on the specific realizations of the D_i variables. In particular, due to the random nature of customers' demands, route failures will occur whenever the aggregated demand in a route exceeds the vehicle capacity. After a route failure, costly corrective actions will need to be considered.

Let $\emptyset \neq S \subset V$ be a subset of nodes, $\delta^+(S)$ the set of arcs leaving S, $\delta^-(S)$ the set of arcs entering S, and $L(S)$ the set of arcs with both ends in S. Our main goal will be to minimize the expected total cost, which can be formulated as:

$$\min \sum_{w \in W} o_w y_w + \sum_{k \in K} \sum_{a \in \delta^+(W)} u f_{ak} + \sum_{k \in K} E[R_k] \tag{1}$$

The term $E[R_k]$ represents the expected value of the following piecewise routing cost function:

$$R_k = \begin{cases} \sum_{a \in A} c_a f_{ak} & \text{if } \sum_{i \in I} \sum_{a \in \delta^-(i)} E[D_i] f_{ak} \leq (1 - \%SS)h \\ \sum_{a \in A} c_a f_{ak} + \rho & \text{otherwise,} \end{cases} \tag{2}$$

where ρ represents the cost of the corrective action required after a route failure:

In the computational experiments carried out in this paper, the cost of each corrective action is computed as $\rho = min\{cost_{reactive}, cost_{preventive}\}$, where *reactive* and *preventive* refer to the following route-repairing strategies:

- *Reactive strategy:* once the vehicle reaches a customer whose demand exceeds its current available load (including the safety stock), it completes a round-trip to the depot facility for a refill.

- *Preventive strategy:* after serving each customer, if the expected value of the non-served demand in a route exceeds the vehicle load (including the safety stock), the vehicle completes a 'detour' via the depot before visiting the next customer.

It is assumed that visiting a depot for a truck reload is always possible after a route failure, since depot facilities are refilled with new products once the route distribution starts. In addition, the following constraints also apply during the solution-design stage:

$$\sum_{k\in K}\sum_{a\in\delta^-(i)} f_{ak} = 1 \qquad \forall i \in I \tag{3}$$

$$\sum_{i\in I}\sum_{a\in\delta^-(i)} E[D_i] f_{ak} \leq (1 - \%SS)h \qquad \forall k \in K \tag{4}$$

$$\sum_{a\in\delta^+(j)} f_{ak} - \sum_{a\in\delta^-(j)} f_{ak} = 0 \qquad \forall k \in K, \forall j \in V \tag{5}$$

$$\sum_{a\in\delta^+(i)} f_{ak} \leq 1 \qquad \forall k \in K, \forall i \in I \tag{6}$$

$$\sum_{a\in L(S)} f_{ak} \leq |S| - 1 \qquad \forall S \subseteq I, \forall k \in K \tag{7}$$

$$\sum_{a\in\delta^+(w)\cap\delta^-(I)} f_{ak} + \sum_{a\in\delta^-(i)} f_{ak} \leq 1 + x_{iw} \qquad \forall i \in I, \forall w \in W, \forall k \in K \tag{8}$$

$$\sum_{i\in I} E[D_i] x_{iw} \leq q_w y_w \qquad \forall w \in W \tag{9}$$

$$f_{ak}, x_{iw}, y_w \in \{0, 1\} \qquad \forall a \in A, \forall k \in K, \forall i \in I, \forall w \in W \tag{10}$$

Constraint (3) ensures that each customer is visited exactly once. Notice that this constraint only applies during the solution-design stage, and it might be violated during the execution stage if a route failure occurs and a repairing action is required. If a reactive strategy is applied, the customer will be visited twice. However, we assume in our formulation that the delivery cannot be split —i.e., the customer is serviced just once—, but we account for the associated cost of such "split" delivery in the objective function (ρ). Expression (4) ensures vehicle capacity is not exceeded —i.e., the expected demand to be serviced by each vehicle cannot exceed its designated loading capacity, excluding the safety stock. Notice that, if safety stocks are not considered (i.e., $\%SS = 0$), this expression turns into a standard capacity constraint. Inequalities (5) and (6) guarantee the continuity of each designed route and the return of a route to its origin depot. Inequalities (7) are subtour elimination constraints. Equations (8) guarantee that a customer is only assigned to a facility if there are routes originating from that facility. Constraint (9) specifies that the expected demand to be serviced from a depot facility cannot exceed its initial capacity. Finally, expressions (10) define the decision variables.

4. Our SimILS Solving Approach

In order to efficiently solve the CLRPSD, we have designed a simheuristic procedure that follows a SimILS framework (Grasas et al., 2016) and also makes use of biased randomization techniques (Dominguez et al., 2016; Grasas et al., 2017). The principles inspiring this method are: *(i)* to design a relatively easy-to-implement algorithm that does not require a complex parameter fine-tuning process —all parameter values have been obtained after a quick trial-and-error process; and, *(ii)* to allow a fast feedback among the different sub-problems that compose the CLRP in order to provide an 'integrated' (non-sequential) approach. Our simheuristic algorithm consists of two main stages (Figure 2). In the first stage, many facility-location maps are quickly generated and tested using fast allocation and routing heuristics. The most 'promising' maps obtained in this first stage are then sent to a second stage, where the customer-to-facility allocation and vehicle routing processes are further improved by using more intensive algorithms. Both stages are explained next in more detail.

4.1. Generation of Feasible Solutions

In order to obtain base facility-location maps, the first step is to estimate the number of facilities required to serve all customers' demands. Hence, we start by computing a lower bound (lb) on the number of facilities to be opened by dividing the expected total demand by the maximum capacity in the set of facilities. Next, $nComb$ combinations of open facilities ($nComb = 30$ in our experiments) are randomly generated for each value of l, with $lb \leq l \leq |W|$. For each of these combinations, the associated CLRP is solved by means of fast allocation and routing heuristics. The average cost is computed for each l, and the l value with the lowest average cost is kept as a 'reasonable' number of facilities to be opened. At this point, the value of lb is updated as the maximum between its original value and $l - 1$. Similarly, the value of the upper bound ub is set to $l + 1$. Then, we generate random sets of l^* open facilities (with $lb \leq l^* \leq ub$), during at most $nIters$ iterations ($nIters = 1000$ in our experiments). On each of these sets, the following allocation process is applied:

- For each facility, customers are sorted in a list of candidates according to the marginal-savings criterion proposed by Juan et al. (2015). This

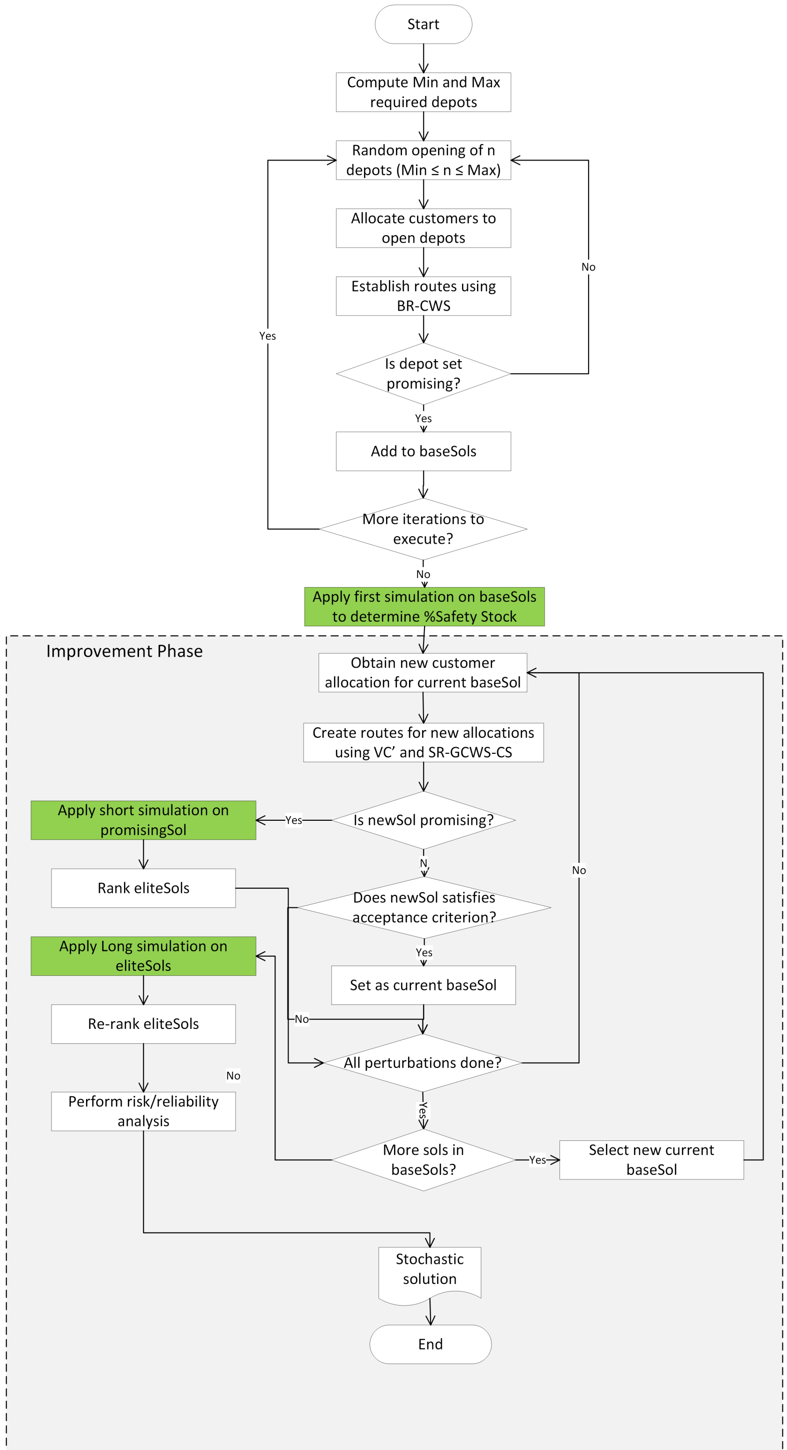

Figure 2: Flowchart of our simheuristic algorithm.

criterion computes the savings of assigning a customer $i \in I$ to an open facility $w \in W$ with respect to assigning i to the best alternative facility $w^* \in W \setminus \{w\}$.

- Next, a diminishing probability of being selected is assigned to each customer; i.e., the higher its position in the sorted list, the higher the probability of being assigned to (or selected by) the corresponding facility. As it is usually the case in biased randomization (Grasas et al., 2017), these probabilities are defined by employing a skewed probability distribution. In our case, we use a geometric distribution since it contains just one parameter, β, which during our experiments is set to take random values in the interval $(0.05, 0.80)$.

- Then, a round-robin selection process is started: the selection turn iterates over the set of facilities, and the facility with the turn is allowed to select a new customer from its sorted list according to the previously established probabilities.

- When all customers have been allocated to a given facility, the fast Clarke and Wright (1964) routing heuristic is applied to generate an initial routing plan.

- Moreover, during the route planning we also adopt the strategy of using safety stocks in the vehicles as a security buffer against higher-than-expected demands. Accordingly, the recommended value of the safety-stock percentage ($\%SS$), which is used as a protection against demand uncertainty, is determined by the first simulation process described in the flowchart. To set it, we run a 'fast simulation (100 iterations) for each instance, varying the %SS from 0 to 10%; then, for each value of the %SS, we compute the results obtained by the simulation runs, and keep the %SS producing the lowest estimated stochastic cost. In addition, this methodology is simple to implement, so it could provide a good first estimate of the required $\%SS$ in practical applications.

Once the initial iterations have been completed, a set of $nBaseSols$ base solutions ($nBaseSols = 2 + \lfloor nodes/100 \rfloor$ in our experiments) are kept to be refined in the improvement stage.

4.2. Solution Improvement

In this stage, the goal is to explore in more detail each of the base solutions so that better customer-to-facility allocation and routing plans can be identified in them. The pseudocode of this stage is presented in Algorithm 1. During this stage, however, the open facilities in each map are not changed. Two alternative procedures are employed. In the first one, *Rand*, the perturbation operator randomly selects a set of customers and tries to reassign them to another facility without violating its capacity. The number of customers that are randomly selected ranges from 2 to the rounded average number of customers in a route —which is instance-dependent. In the second perturbation procedure the perturbation operator randomly exchanges the allocation of $p\%$ of the customers. In our experiments, $p \in \{0.05, 0.1, \ldots, 0.95\}$; i.e., we start with the lowest value of p, which is successively increased to explore different neighborhood sizes. In both versions of the algorithm, after each new customer-to-facility allocation, a higher-quality routing process is executed. In our case, this routing process is carried out by the SR-GCWS-CS biased-randomized algorithm proposed by Juan et al. (2011), which also uses a geometric distribution with parameter α —in our experiments, α is randomly selected in the $(0.07, 0.23)$ interval.

For each new solution, a local search procedure is also applied. As a way to explore a richer neighborhood structure, our algorithm implements four local search operators (Figure 3). In the *Customer Swap Inter-Route* operator, customers are randomly chosen from different routes belonging to the same facility and then swapped. The *Inter-Depot Node Exchange* operator exchanges two nodes randomly selected from different facilities. The *2-Opt Inter-Route* operator interchanges two chains of randomly selected customers between different facilities. Finally, the *Cross-Exchange* operator interchanges positions of 3 randomly selected and non-consecutive customers from different facilities.

Once the local search process is completed, we quickly assess the obtained solution under stochastic conditions by means of a limited MCS process (500 runs, in our experiments). Whenever a new solution outperforms the current base solution of the iterated local search, the latter is updated with the former. Moreover, we also apply two different acceptance criteria to reduce the chances of getting trapped in a local minimum during the search process: the first one is a Demon-based criterion, in which we accept a non-improving solution if its cost is lower or equal to 1.5 times the cost of the base solution; the second one is a simulated annealing (SA) criterion (Henderson et al., 2003)

Algorithm 1: Solution Improvement

```
Input: B, α                    // Set of Base Solutions, Parameter for
  biased-randomized version of CWS
M ← ∅                          // Set of Promising Solutions
for each baseSol ∈ B do
    while stopping criterion not reached do
        newMap ← perturbate map (baseSol)      // Using Rand or p%
        newSol ← route(newMap)                 // Using SR-GCWS-CS
        improving ← true                       // Start Local Search
        while improving do
            newSol* ← localSearch(newSol, LSoperator)
            if costs(newSol*) ≤ costs(newSol) then
                newSol ← newSol*
                add newSol to M
            end
            else
                improving ← false
                if acceptance criterion is met     // Demon or SA
                 then
                    baseSol ← newSol
                end
            end
        end
    end
end
return M
```

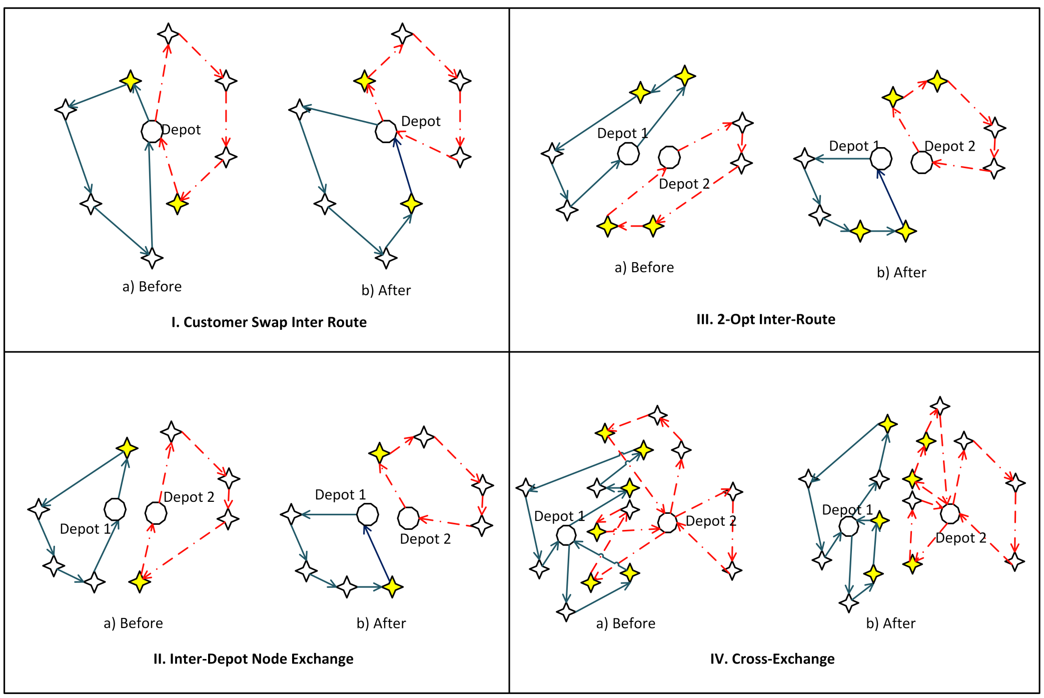

Figure 3: Local search operators.

for non-improving solutions based on an initial temperature t_0 and a cooling factor cf (in our experiments, $t_0 = 100$ and $cf = 0.994$). Finally, a list containing the most promising 10 solutions found so far is updated if necessary. This procedure is repeated until a specified number of iterations ($maxIter$) is reached. Finally, each of the elite solutions obtained goes through a more intensive simulation process with $longSim$ runs ($longSim = 5000$ in our experiments), which allows to obtain better estimates for the total expected cost and the reliability index.

The reliability $reliab_i$ of each route i in solution S is calculated as follows:

$$reliab_i = (1 - \frac{\sum_{n=0}^{longSim} RouteFailuresCount}{longSim}) * 100\% \tag{11}$$

Notice that each route in a solution can be seen as an independent component of a series system —i.e., the proposed solution will fail if, and only if, a failure occurs in any of its routes. Thus, the reliability index of a solution S with R routes can be computed as $\prod_{i=1}^{R} reliab_i$.

The procedure used to evaluate the quality of the solutions under stochastic settings is depicted in Algorithm 2

5. Numerical Experiments and Results

The proposed SimILS algorithm has been coded as a Java application and tested using a standard PC with a Core i5 @2.4GHz CPU and 8Gb RAM. For comparison purposes, four different versions of our solving approach have been considered (see Section 4.2 for a detailed description of each version): $Rand+Demon$, $Rand+SA$, $p\%+Demon$, and $p\%+SA$. The algorithm has been applied to solve both deterministic CLRP instances and the generated CLRPSD ones. For the CLRP, three classical benchmark sets have been used:

- *Akca's set*, introduced by Akca et al. (2009), includes 12 instances with 5 depot facilities and 30-40 customers.
- *Barreto's set*, proposed by Barreto (2004), contains a total of 17 instances with 2 to 15 possible facility locations and 12 to 150 customers.
- *Prodhon's set*, proposed by Belenguer et al. (2011), includes a total of 30 instances ranging from 5 to 10 potential facilities and 20-200 customers.

Algorithm 2: Simulation of stochastic demands

Input: M, $shortSim$, $longSim$, $Var[d_i]$; `// Set of promising deterministic solutions, short and long simulation runs, and demand variance level`

E $\leftarrow \emptyset$; `// Set of Elite Solutions`

for *each solution* $\in M$ **do**

    run short simulation ($shortSim$)

    estimate $expectedStochCosts$

    if *solution among best stochastic solutions* **then**

        include solution in E

    end

end

for *each solution* $\in E$ **do**

    run long simulation ($longSim$)

end

refine $expectedStochCosts$

estimate $realiabilityIndex$

return Set of elite solutions for the stochastic problem

5.1. CLRP with Deterministic Demands

In order to assess the competitiveness of our algorithm in the deterministic scenario, we have tested it on each of the aforementioned benchmark sets. Tables 1, 2 and 3 show the results obtained for Akca's, Barreto's, and Prodhon's sets, respectively, after 10 executions of our algorithm —each execution using a different seed for the pseudo-random number generator. For each instance, the respective table provides the best-known solution (BKS) in the literature, as well as the best and average gaps with respect to each variant of our algorithm. The last row shows averages across all instances. Each BKS corresponds to the best solution among the ones reported in the following series of articles: Yu et al. (2010), Hemmelmayr et al. (2012), Ting and Chen (2013), Contardo et al. (2014), Escobar et al. (2014), and Quintero-Araujo et al. (2017a). Notice that all variants of the algorithm provide competitive results for these sets. In the case of Akca's set (Table 1), average gaps of our best solutions range from 0.01% ($Rand$+SA) to 0.11% ($Rand$+$Demon$). Regarding Barreto's set (Table 2), average gaps of our best solutions vary from 0.45% ($Rand+Demon$) to 0.96% ($p\%+SA$). Finally, for Prodhon's set (Table 3), average gaps vary from 0.50% ($Rand$+$Demon$) to 0.64% ($Rand$+SA). Regarding the gaps for the average of the best solutions reported for each of the 10 runs, they vary from 0.36%($Rand + Demon$) to 0.56%($p\% + SA$) for Akca's set, from 1.15%($Rand + Demon$) to 1.56%($p\% + SA$) for Barreto's set, and 1.19%($Rand + Demon$) to 1.32%($p\% + Demon$) for Prodhon's set.

5.2. CLRP with Stochastic Demands

In order to test our approach in the CLRPSD, we have extended the classical benchmark sets so they can be used in a scenario under uncertainty. Thus, if the original deterministic demand for customer i is given by $d_i > 0$, the transformed demand is a random variable $D_i \geq 0$ which follows a log-normal probability distribution with $E[D_i] = d_i$. Also, in order to consider different levels of uncertainty we have defined $Var[D_i] = \lambda \cdot d_i$, with $\lambda \in \{0.05, 0.1, 0.2\}$ —low, medium, and high variance, respectively. Notice that the log-normal distribution has been selected for running our experiments since it is a flexible distribution frequently used to model non-negative random variables. In a real-life application, however, a best-fit analysis would need to be carried out to decide the specific distribution to be used.

For every extended benchmark set and uncertainty level, Tables 4 to 9 provide the results obtained with our algorithm. As in the deterministic case, 10 runs per instance were executed for each variant of the algorithm.

Table 1: Comparison of our algorithm against the best-known solution for the deterministic CLRP - Akca's set.

		% Gap with respect to the BKS							
Instance	BKS	Rand+Demon Best	Rand+Demon Average	Rand+SA Best	Rand+SA Average	p%+Demon Best	p%+Demon Average	p%+SA Best	p%+SA Average
cr30x5a-1	819.51	0.00	0.02	0.00	0.04	0.00	0.04	0.00	0.27
cr30x5a-2	821.45	0.00	0.45	0.00	0.45	0.00	0.52	0.07	0.58
cr30x5a-3	702.29	0.00	0.74	0.00	0.44	0.00	0.73	0.00	0.66
cr30x5b-1	880.02	0.00	0.24	0.00	0.79	0.00	0.20	0.00	0.54
cr30x5b-2	825.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
cr30x5b-3	884.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
cr40x5a-1	928.10	0.37	0.57	0.16	0.54	0.17	0.57	0.37	0.64
cr40x5a-2	888.42	0.04	0.21	0.00	0.09	0.00	0.14	0.00	0.16
cr40x5a-3	947.26	0.00	0.34	0.00	0.57	0.00	0.48	0.00	0.89
cr40x5b-1	1052.04	0.00	0.43	0.00	0.42	0.00	0.53	0.00	0.73
cr40x5b-2	981.54	0.92	1.21	0.00	1.13	0.00	1.15	0.00	1.16
cr40x5b-3	964.33	0.00	0.17	0.00	0.43	0.00	0.53	0.00	1.09
Averages		0.11	0.36	0.01	0.41	0.01	0.41	0.04	0.56

Table 2: Comparison of our algorithm against the best-known solution for the deterministic CLRP - Barreto's set.

		% Gap with respect to the BKS							
Instance	BKS	Rand+Demon Best	Rand+Demon Average	Rand+SA Best	Rand+SA Average	p%+Demon Best	p%+Demon Average	p%+SA Best	p%+SA Average
Perl83-12x2	203.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Gaskell-21x5	424.90	0.00	0.72	0.00	0.65	0.00	0.60	0.00	0.40
Gaskell-22x5	585.11	0.00	0.78	0.00	0.78	0.00	0.78	0.00	0.78
Min-27x5	3062.02	0.00	0.02	0.00	0.02	0.00	0.01	0.00	0.01
Gaskell-29x5	512.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Gaskell-32x5	562.22	0.01	0.50	0.01	0.50	0.01	0.50	0.01	0.50
Gaskell-32x5B	504.33	0.00	0.60	0.00	0.60	0.00	0.60	0.00	0.60
Gaskell-36x5	460.37	3.39	4.40	3.39	4.40	3.39	4.40	3.39	4.40
Christ-50x5	565.62	0.00	1.23	1.16	2.05	1.39	1.97	1.42	2.36
Christ-50x5_B	565.60	0.00	1.92	1.20	2.99	0.00	2.85	2.06	3.60
Perl83-55x15	1112.06	0.20	0.67	0.79	1.04	0.41	0.87	0.79	1.04
Christ-75x10_B	844.40	0.68	1.35	1.03	1.78	0.98	1.73	1.03	1.78
Perl-85x7	1622.50	0.33	0.90	0.83	1.22	0.78	1.14	1.12	1.32
Daskin95-88x8	355.78	0.09	0.85	0.11	1.35	0.06	0.84	0.07	1.09
Christ-100x10	833.43	1.36	2.25	1.47	3.22	1.35	3.03	3.00	3.80
Min92-134x8	5709.00	1.59	2.28	1.52	2.77	2.02	3.22	2.02	3.21
Daskin95-150x10	43919.90	0.09	1.10	0.47	1.12	0.94	1.48	1.32	1.59
Averages		0.45	1.15	0.71	1.44	0.67	1.41	0.96	1.56

Table 3: Comparison of our algorithm against the best-known solution for the deterministic CLRP - Prodhon's set.

		% Gap with respect to the BKS							
Instance	BKS	Rand+Demon Best	Rand+Demon Average	Rand+SA Best	Rand+SA Average	p%+Demon Best	p%+Demon Average	p%+SA Best	p%+SA Average
coord100-10-1	287695	1.12	2.14	1.12	2.14	1.04	2.11	1.04	2.11
coord100-10-1b	230989	0.97	1.62	0.97	1.62	0.97	1.64	0.97	1.64
coord100-10-2	243590	0.10	0.72	0.15	0.90	0.15	0.91	0.10	0.78
coord100-10-2b	203988	0.52	0.84	0.61	1.01	0.52	1.14	0.61	0.91
coord100-10-3	250882	1.29	2.69	1.64	2.92	2.17	3.07	1.33	2.77
coord100-10-3b	204317	0.69	2.09	0.90	2.27	0.59	2.48	0.53	2.25
coord100-5-1	274814	0.83	1.80	1.14	1.96	0.77	1.87	0.77	1.87
coord100-5-1b	213615	0.68	1.29	0.58	1.32	0.87	1.60	0.87	1.51
coord100-5-2	193671	0.90	1.36	0.90	1.38	0.90	1.41	0.90	1.41
coord100-5-2b	157095	0.14	0.46	0.14	0.51	0.25	0.52	0.25	0.56
coord100-5-3	200079	0.84	1.48	0.84	1.45	0.84	1.48	0.84	1.45
coord100-5-3b	152441	0.00	0.88	0.18	0.95	0.08	0.92	0.02	0.83
coord20-5-1	54793	0.00	0.40	0.00	0.40	0.42	0.42	0.42	0.42
coord20-5-1b	39104	0.00	1.81	0.00	1.81	0.00	1.81	0.00	1.81
coord20-5-2	48908	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
coord20-5-2b	37542	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
coord200-10-1	475294	0.60	1.03	0.84	1.11	0.71	0.99	0.74	0.96
coord200-10-1b	377043	0.19	0.60	0.52	0.79	0.33	0.54	0.33	0.52
coord200-10-2	449006	0.40	0.51	0.44	0.55	0.32	0.50	0.33	0.49
coord200-10-2b	374280	0.22	0.46	0.33	0.48	0.18	0.39	0.16	0.41
coord200-10-3	469433	0.98	1.33	1.06	1.37	1.04	1.34	1.02	1.33
coord200-10-3b	362653	1.72	2.47	1.85	2.58	1.60	2.34	1.49	2.25
coord50-5-1	90111	0.00	0.20	0.00	0.20	0.00	0.45	0.00	0.35
coord50-5-1b	63242	0.00	0.36	0.00	0.36	0.00	0.38	0.00	0.38
coord50-5-2	88298	0.00	2.15	1.81	2.14	1.74	2.51	1.30	2.38
coord50-5-2b	67308	0.79	1.96	0.79	2.16	0.79	2.88	1.27	2.46
coord50-5-2bBIS	51822	0.12	0.85	0.12	0.91	0.26	0.93	0.26	0.97
coord50-5-2BIS	84055	0.91	2.18	1.30	2.42	1.12	2.84	1.12	2.42
coord50-5-3	86203	0.81	1.00	0.81	1.01	0.81	1.04	0.81	1.00
coord50-5-3b	61830	0.02	0.92	0.00	0.95	0.00	1.04	0.00	0.90
Averages		0.50	1.19	0.64	1.26	0.62	1.32	0.58	1.24

Table 4: CLRPSD - Results with low variance level - Prodhon's set.

Prodhon's set			
Instance Name	OBS	Reliability OBS	Average BS
coord100-10-1	294626.39	65%	298047.40
coord100-10-1b	236796.93	91%	238549.54
coord100-10-2	245473.87	44%	246966.81
coord100-10-2b	205216.71	96%	205898.37
coord100-10-3	255349.96	65%	258758.80
coord100-10-3b	204979.01	89%	208624.85
coord100-5-1	279719.50	45%	282326.53
coord100-5-1b	214935.44	88%	216428.59
coord100-5-2	196738.12	36%	197925.23
coord100-5-2b	157470.09	89%	158186.99
coord100-5-3	203498.59	47%	205249.86
coord100-5-3b	152747.22	87%	153811.03
coord200-10-1	481720.03	34%	484073.75
coord200-10-1b	378394.52	85%	379109.76
coord200-10-2	451637.49	45%	452750.55
coord200-10-2b	374801.79	78%	375533.78
coord200-10-3	477448.21	29%	479216.40
coord200-10-3b	368657.09	88%	371277.35
coord20-5-1	55247.55	79%	55365.87
coord20-5-1b	39104.00	100%	39734.30
coord20-5-2	48909.34	100%	48912.14
coord20-5-2b	37542.00	100%	37542.20
coord50-5-1	90264.79	97%	90597.48
coord50-5-1b	63351.49	93%	63642.68
coord50-5-2	89288.34	62%	90247.94
coord50-5-2b	68337.99	91%	68738.73
coord50-5-2bBIS	51940.11	99%	52335.05
coord50-5-2BIS	85237.35	96%	86540.19
coord50-5-3	86556.65	87%	87841.87
coord50-5-3b	61856.79	99%	62492.06
Average		77%	

Table 5: CLRPSD - Results with low variance level - Akca's and Barreto's sets.

Akca's set			
Instance Name	OBS	Reliability OBS	Average BS
cr30x5a-1	821.61	94%	823.62
cr30x5a-2	821.45	100%	840.12
cr30x5a-3	706.82	85%	715.54
cr30x5b-1	881.14	98%	887.43
cr30x5b-2	825.32	100%	825.32
cr30x5b-3	886.32	96%	887.88
cr40x5a-1	929.16	100%	934.27
cr40x5a-2	888.80	100%	892.54
cr40x5a-3	953.31	93%	956.84
cr40x5b-1	1057.29	86%	1063.49
cr40x5b-2	981.59	100%	994.60
cr40x5b-3	972.10	92%	994.60
Average		95%	
Barreto's set			
Instance Name	OBS	Reliability OBS	Average BS
Christ-100x10	847.75	98%	858.19
Christ-50x5	566.59	95%	577.00
Christ-50x5_B	566.51	97%	585.83
Christ-75x10	814.09	97%	825.27
Daskin95-150x10	44109.22	100%	44647.12
Daskin95-88x8	358.46	100%	361.15
Gaskell-21x5	427.02	89%	430.26
Gaskell-22x5	585.11	100%	585.11
Gaskell-29x5	512.10	100%	512.10
Gaskell-32x5	562.28	100%	562.28
Gaskell-32x5-2	504.33	100%	504.73
Gaskell-36x5	460.37	100%	467.99
Min-27x5	3062.02	100%	3063.04
Min92-134x8	5775.24	83%	5855.16
Perl83-12x2	205.11	91%	205.23
Perl83-55x15	1131.99	33%	1141.42
Average		93%	

Table 6: CLRPSD - Results with medium variance level - Prodhon's set.

Prodhon's set			
Instance Name	OBS	Reliability OBS	Average BS
coord100-10-1	297607.12	43%	300028.21
coord100-10-1b	237622.33	70%	239227.00
coord100-10-2	247557.24	53%	248679.66
coord100-10-2b	205346.71	85%	206174.83
coord100-10-3	256562.35	39%	260163.32
coord100-10-3b	205140.82	95%	208880.07
coord100-5-1	283749.90	41%	285438.82
coord100-5-1b	215697.97	67%	217237.49
coord100-5-2	198599.72	50%	199418.38
coord100-5-2b	157849.70	66%	158710.05
coord100-5-3	204990.10	23%	207722.84
coord100-5-3b	153547.46	60%	154497.86
coord200-10-1	486959.87	14%	488476.75
coord200-10-1b	379549.65	54%	380307.91
coord200-10-2	454426.38	28%	455122.16
coord200-10-2b	375421.53	64%	376295.25
coord200-10-3	480682.29	17%	482799.15
coord200-10-3b	369208.51	78%	372648.54
coord20-5-1	55516.43	77%	55616.44
coord20-5-1b	39104.00	100%	39750.63
coord20-5-2	48957.36	97%	48968.73
coord20-5-2b	37547.58	100%	37555.71
coord50-5-1	90781.90	83%	91297.16
coord50-5-1b	63685.50	79%	64014.31
coord50-5-2	90211.48	67%	90824.84
coord50-5-2b	68632.11	76%	68958.79
coord50-5-2bBIS	52085.86	95%	52403.54
coord50-5-2BIS	85990.77	80%	87650.10
coord50-5-3	87089.14	67%	88526.45
coord50-5-3b	61920.53	100%	62647.16
Average		63%	

Table 7: CLRPSD - Results with medium variance level - Akca's and Barreto's set.

Akca's set			
Instance Name	OBS	Reliability OBS	Average BS
cr30x5a-1	825.92	84%	829.83
cr30x5a-2	821.73	100%	839.13
cr30x5a-3	709.73	76%	722.64
cr30x5b-1	885.58	95%	892.31
cr30x5b-2	825.32	95%	825.35
cr30x5b-3	891.07	86%	899.05
cr40x5a-1	932.54	98%	935.74
cr40x5a-2	888.82	100%	895.68
cr40x5a-3	957.14	93%	961.14
cr40x5b-1	1064.10	98%	1069.48
cr40x5b-2	983.60	98%	999.85
cr40x5b-3	974.94	82%	988.03
Average		92%	
Barreto's set			
Instance Name	OBS	Reliability OBS	Average BS
Christ-100x10	850.60	92%	860.10
Christ-50x5	569.57	82%	579.64
Christ-50x5_B	571.01	98%	587.43
Christ-75x10	815.04	95%	828.04
Daskin95-150x10	44161.74	100%	44677.73
Daskin95-88x8	358.46	100%	361.20
Gaskell-21x5	429.31	78%	432.24
Gaskell-22x5	585.11	100%	585.11
Gaskell-29x5	512.10	100%	512.10
Gaskell-32x5	562.28	100%	562.28
Gaskell-32x5-2	504.33	100%	504.73
Gaskell-36x5	460.37	100%	467.99
Min-27x5	3062.02	100%	3062.64
Min92-134x8	5781.37	94%	5867.06
Perl83-12x2	206.07	83%	206.32
Perl83-55x15	1142.09	100%	1154.21
Average		90%	

Table 8: CLRPSD - Results with high variance level - Prodhon's set.

Prodhon's set			
Instance Name	OBS	Reliability OBS	Average BS
coord100-10-1	300916.95	34%	303196.53
coord100-10-1b	238526.38	83%	240545.39
coord100-10-2	249105.62	65%	250984.54
coord100-10-2b	205626.26	64%	207276.45
coord100-10-3	258139.10	35%	262784.79
coord100-10-3b	205602.99	76%	210441.76
coord100-5-1	287244.95	42%	290747.84
coord100-5-1b	216896.88	77%	219040.91
coord100-5-2	200123.80	52%	201997.61
coord100-5-2b	158467.00	61%	159804.59
coord100-5-3	208037.01	27%	210916.70
coord100-5-3b	154944.33	56%	156050.25
coord200-10-1	492303.88	21%	495255.29
coord200-10-1b	380938.57	46%	383186.23
coord200-10-2	458041.13	18%	459786.11
coord200-10-2b	377080.38	62%	378341.81
coord200-10-3	485618.47	17%	489186.88
coord200-10-3b	371555.87	45%	376079.51
coord20-5-1	55734.66	69%	56021.61
coord20-5-1b	39108.36	100%	39787.08
coord20-5-2	49181.15	87%	49204.32
coord20-5-2b	37633.17	97%	37650.04
coord50-5-1	91776.95	52%	92527.43
coord50-5-1b	64127.77	77%	64890.71
coord50-5-2	91555.55	45%	92282.31
coord50-5-2b	68743.07	94%	69508.43
coord50-5-2bBIS	52167.69	95%	52709.41
coord50-5-2BIS	88034.97	48%	90037.87
coord50-5-3	88004.55	65%	89730.70
coord50-5-3b	62129.90	86%	63020.86
Average		58%	

Table 9: CLRPSD - Results with high variance level - Akca's and Barreto's sets.

Akca's set			
Instance Name	OBS	Reliability OBS	Average BS
cr30x5a-1	834.50	67%	842.36
cr30x5a-2	824.80	96%	844.54
cr30x5a-3	713.60	66%	727.94
cr30x5b-1	890.97	86%	901.08
cr30x5b-2	825.36	100%	825.46
cr30x5b-3	901.21	68%	919.42
cr40x5a-1	933.88	87%	940.62
cr40x5a-2	889.60	99%	898.82
cr40x5a-3	962.63	92%	967.14
cr40x5b-1	1071.89	89%	1079.06
cr40x5b-2	991.09	88%	1011.41
cr40x5b-3	980.73	69%	993.07
Average		84%	
Barreto's set			
Instance Name	OBS	Reliability OBS	Average BS
Christ-100x10	858.42	82%	865.80
Christ-50x5	574.55	62%	583.31
Christ-50x5_B	574.23	87%	590.07
Christ-75x10	821.27	85%	835.44
Daskin95-150x10	44222.57	100%	44636.13
Daskin95-88x8	358.46	100%	361.19
Gaskell-21x5	432.29	67%	435.38
Gaskell-22x5	585.11	100%	585.11
Gaskell-29x5	512.10	100%	512.10
Gaskell-32x5	562.28	100%	562.30
Gaskell-32x5-2	504.33	100%	504.76
Gaskell-36x5	460.49	100%	468.20
Min-27x5	3062.03	100%	3063.14
Min92-134x8	5802.19	74%	5889.38
Perl83-12x2	207.27	73%	207.42
Perl83-55x15	1145.02	100%	1165.06
Average		89%	

Table 10: CLRPSD - Comparison among variants - low variance level.

	GAP w.r.t. OBS			
	rand + DEMON	*rand + SA*	*p% + DEMON*	*p% + SA*
INSTANCE NAME	**PRODHON'S SET**			
coord100-10-1	0.00%	0.28%	0.21%	0.17%
coord100-10-2b	0.01%	0.01%	0.00%	0.16%
coord100-10-3	0.07%	0.00%	0.50%	0.16%
coord100-10-3b	0.18%	0.00%	0.35%	0.38%
coord100-5-1	0.13%	0.06%	0.15%	0.00%
coord100-5-2	0.09%	0.00%	0.12%	0.03%
coord100-5-2b	0.15%	0.15%	0.00%	0.01%
coord100-5-3	0.00%	0.00%	0.22%	0.22%
coord100-5-3b	0.00%	0.03%	0.02%	0.02%
coord200-10-1	0.15%	0.00%	0.18%	0.27%
coord200-10-1b	0.10%	0.06%	0.00%	0.01%
coord200-10-2	0.04%	0.04%	0.06%	0.00%
coord200-10-2b	0.03%	0.08%	0.00%	0.06%
coord200-10-3	0.08%	0.00%	0.19%	0.07%
coord200-10-3b	0.20%	0.10%	0.05%	0.00%
coord20-5-1	0.00%	0.00%	0.13%	0.11%
coord20-5-1b	0.00%	0.00%	0.00%	0.00%
coord20-5-2	0.00%	0.00%	0.00%	0.00%
coord50-5-1	0.03%	0.02%	0.00%	0.00%
coord50-5-1b	0.07%	0.07%	0.00%	0.00%
coord50-5-2	0.00%	0.00%	0.80%	1.12%
coord50-5-2b	0.00%	0.01%	0.04%	0.15%
coord50-5-2bBIS	0.01%	0.00%	0.01%	0.32%
coord50-5-2BIS	0.16%	0.15%	0.00%	0.40%
coord50-5-3	0.60%	0.63%	0.23%	0.00%
coord50-5-3b	0.13%	0.43%	0.00%	0.12%
AVERAGE	0.08%	0.09%	0.13%	0.16%
NUMBER OF BS	10	13	6	6
AKCA'S SET				
cr30x5a-1	0.00%	2.14%	2.25%	2.50%
cr30x5a-2	0.00%	0.00%	0.00%	0.00%
cr30x5a-3	0.04%	0.00%	0.03%	0.03%
cr30x5b-1	0.00%	0.01%	0.01%	0.25%
cr30x5b-2	0.00%	0.00%	0.00%	0.00%
cr30x5b-3	0.01%	0.00%	0.00%	0.01%
cr40x5a-1	0.34%	0.56%	0.00%	0.11%
cr40x5a-2	0.00%	0.00%	0.26%	0.26%
cr40x5a-3	0.00%	0.00%	0.14%	0.13%
cr40x5b-1	0.00%	0.18%	0.18%	0.00%
cr40x5b-2	0.00%	0.00%	0.01%	0.01%
cr40x5b-3	0.61%	2.12%	0.00%	0.68%
AVERAGE	0.08%	0.42%	0.24%	0.33%
NUMBER OF BS	6	6	5	3
BARRETO'S SET				
Christ-100x10	0.40%	0.00%	0.43%	0.47%
Christ-50x5	0.03%	0.68%	0.00%	1.36%
Christ-50x5_B	0.00%	0.69%	1.04%	2.19%
Christ-75x10	0.00%	0.72%	0.35%	0.35%
Daskin95-150x10	0.57%	0.46%	0.80%	0.00%
Daskin95-88x8	0.00%	0.00%	0.00%	0.01%
Gaskell-21x5	0.04%	0.00%	0.05%	0.05%
Gaskell-22x5	0.00%	0.00%	0.00%	0.00%
Gaskell-29x5	0.00%	0.00%	0.00%	0.00%
Gaskell-32x5	0.00%	0.00%	0.00%	0.00%
Gaskell-32x5-2	0.00%	0.00%	0.00%	0.00%
Gaskell-36x5	0.00%	0.00%	0.00%	0.00%
Min-27x5	0.00%	0.00%	0.00%	0.00%
Min92-134x8	0.06%	0.00%	0.06%	0.05%
Perl83-12x2	0.00%	0.00%	0.00%	0.00%
Perl83-55x15	0.00%	0.06%	0.19%	0.46%
AVERAGE	0.07%	0.16%	0.18%	0.31%
NUMBER OF BS	11	11	9	8
AVERAGE	**0.08%**	**0.18%**	**0.17%**	**0.24%**
TOTAL BS	**27**	**30**	**20**	**17**

Each table shows: the best-found solution for each instance of the CLRPSD (OBS) across the four algorithm variants; its expected reliability (Reliability OBS), which is the probability that the designed distribution plan can be executed without route failures; and, the average cost obtained across all runs (Average BS). It is important to mention that, for the same instance with different variability levels, the recommended %SS determined via simulation might vary; e.g., in the case of instance *Coord100-10-1*, the recommended safety stock percentages are 0.0%, 0.0%, and 2.0% for low, medium, and high variability levels, respectively.

Table 10 provides a comparison among the 4 versions of our algorithm for a low variance scenario. For each instance, it compares the results provided by each algorithm variant against our best stochastic solution across methods. Notice that all variants provide solutions of similar quality, both in terms of expected stochastic costs and expected reliability . The $rand + SA$ method is the one that provides the highest number of best solutions (30 out of 59 instances), while $rand + DEMON$ is the version providing the lowest average gaps for all sets. On the other hand, the $p\% + SA$ version seems to have the poorest performance in terms of both average gap and number of best solutions obtained. Our strategy of using simulation to determine the "ideal" value of the safety stock percentage seems to work well, since expected total costs are not too different from those of the deterministic solutions. Also, for the low-variability level scenario, we observe that the average reliability for Akca's and Barreto's instances is over 90%. In fact, except for 3 instances in each set, reliability stays over 90% in all cases. Comparatively, in the case of Prodhon's instances, reliability is below 80%. Regarding these results for Prodhon's set, it is important to remember that our goal is to minimize total expected cost associated with the distribution plan, and not explicitly reliability. Sometimes, it might pay off to accept a lower reliability value if that implies using less vehicles, especially if fixed costs per vehicle are significant. This is the case in Prodhon's set, which explains the low reliability levels obtained for these instances.

To illustrate the relationship between the total expected cost and reliability, we have solved a particular instance (*Cr30x5a-2*) using different safety stock policies. To do so, we have used the algorithm that provides our best solution for this instance under the high variance setting ($rand + Demon$). The behaviors of stochastic costs and reliability are shown in Figure 4. It can be seen that, without safety stocks, the expected costs are high, while the associated reliability is around 60%. Using a 2% safety stock level provides the best solution in terms of expected stochastic costs. In addition, the associated reliability is 96%. Using a 10% safety stock level raises reliability up to almost 100%, but it also increases the total expected cost due to the additional routes and vehicles required.

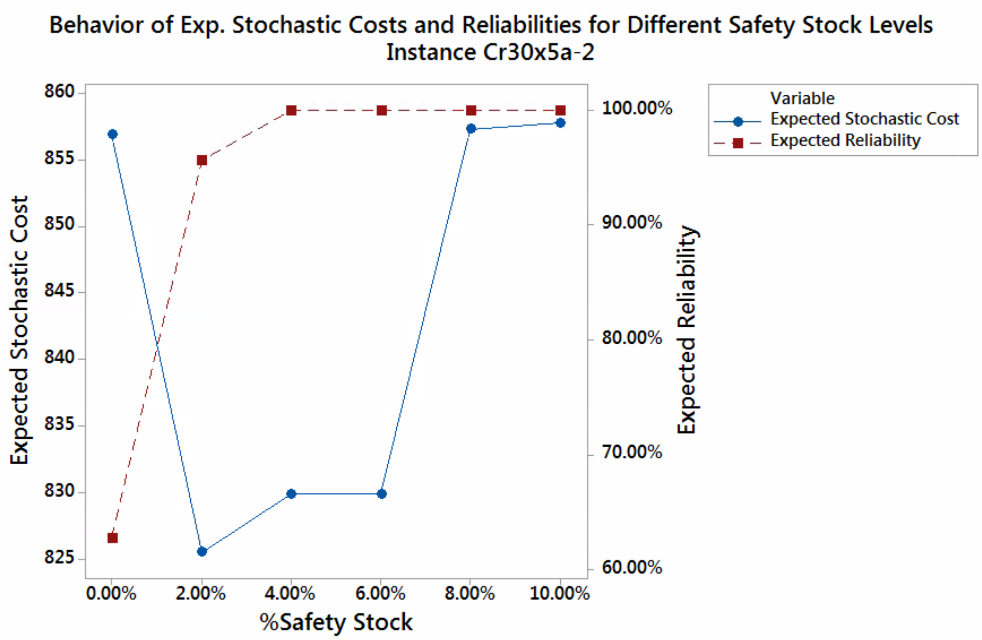

Figure 4: Behavior of costs and reliability with different safety stock policies.

For the same instance, Figure 5 shows the variability of solutions obtained with different safety stock policies when using the *rand* + *Demon* variant of the algorithm. As can be seen, the solution for a 2% safety stock level is better —in terms of total expected cost— than the solution obtained with 4% safety stock level. However, the latter has less variability —i.e., it is a low-risk policy. Thus, a risk-averse decision maker would prefer the latter policy, while a risk-oriented one would select the former.

To show the quality of our simheuristic method, we have compared the best deterministic solution found (BDS), tested in a stochastic setting with high variability, against our best stochastic solution (BSS). To do so, we use six different instances from the different datasets, and we compare the results obtained in terms of both expected stochastic costs and expected reliability. Table 11 provides a cost comparison, while Figure 6 shows a graphical comparison in terms of reliability. According to these results, it is clear that the

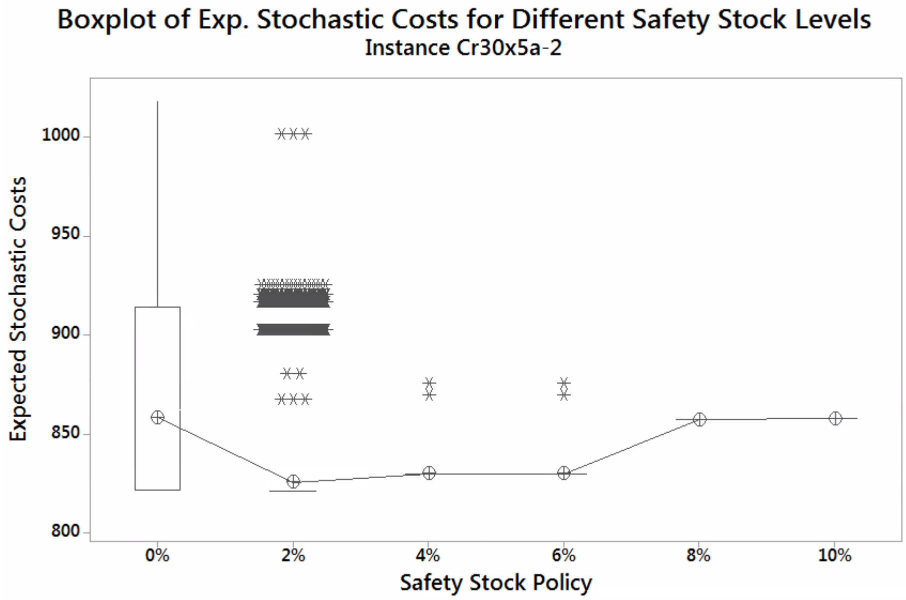

Figure 5: Comparison of best stochastic solutions for different safety stock levels.

solutions obtained by means of our simheuristic approach outperform the deterministic ones when deployed in a scenario with uncertainty. Regarding instances 1 and 5, the number of used vehicles is the same for BSS and BDS, while for instances 2 and 4, the stochastic solution uses one more vehicle than the corresponding BDS. In both situations, the BSS provides slightly higher routing costs, which are then compensated by much lower route failures and, therefore, less stochastic costs and higher reliability levels.

Finally, for instance *coord100-5-1*, Figure 7 shows a multiple boxplot of total costs for different solutions: the best deterministic solution (BDS) when applied in a stochastic scenario with high variability, and the two stochastic solutions with the lowest expected total costs provided by our algorithm (BSS1 and BSS2) for the same scenario. Again, notice that our simheuristic algorithm is able to provide better results in terms of expected total costs as well as variability.

6. Conclusions

The present work analyzes the capacitated location routing problem with stochastic demands, which can cause route failures leading to costly cor-

Table 11: Comparison of expected stochastic costs among BDS and BSS under stochastic settings.

Instance	**Name**	**BDS Expected Costs (1)**	**BSS Expected Cost (2)**	**Gap in Costs (2) - (1)**
1	Coord200-10-2b	378409.86	377080.38	-0.35%
2	Perl83-55x15	1150.84	1139.42	-0.99%
3	Perl83-12x2	207.28	207.27	0.00%
4	Cr40x5b-3	992.83	980.73	-1.22%
5	Coord100-5-1	287689.72	287244.95	-0.15%
6	Cr30x5b-2	825.39	825.36	0.00%
Average				*-0.45%*

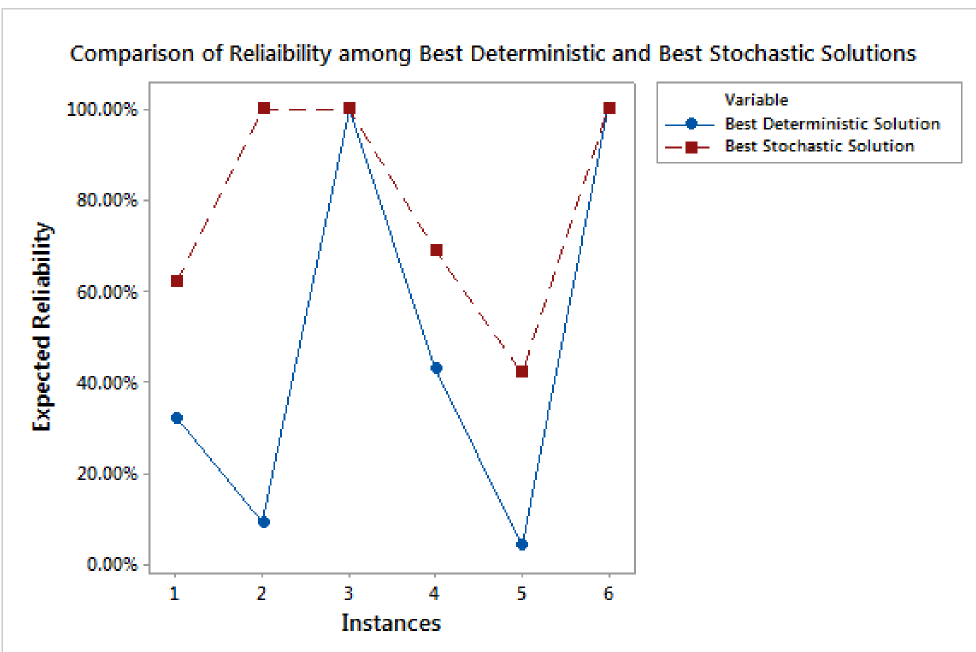

Figure 6: Comparison of reliability among BDS and BSS.

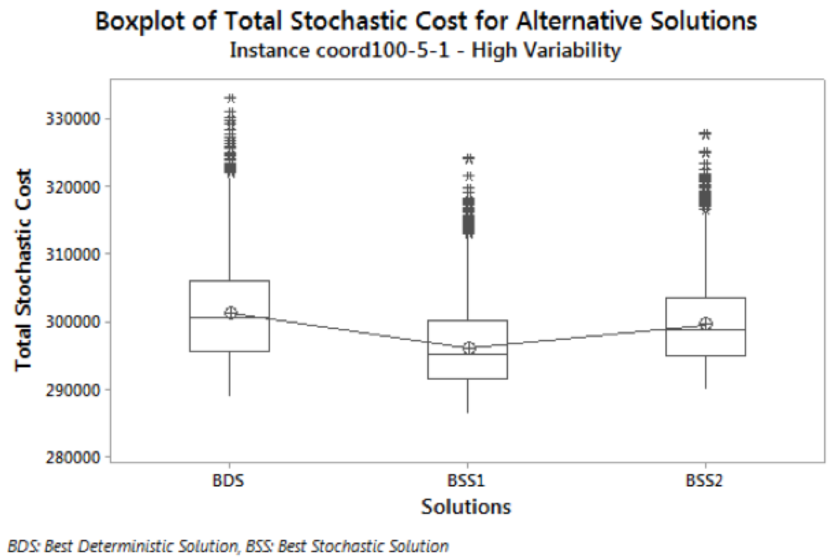

Figure 7: Comparison between deterministic and stochastic solutions for instance *coord100-5-1* with high variability.

rective actions. In this paper, customers' demands are assumed to follow independent log-normal probability distributions with different variability levels. We have proposed four different versions of a simheuristic algorithm to solve the aforementioned problem. Our algorithm combines an iterated local search metaheuristic with Monte Carlo simulation. In addition, it also makes use of biased randomization techniques. Four variants using two different perturbation operators and two different acceptance criteria have also been tested. Among them, the variant combining random selection of customers and a Demon-based acceptance criterion ($rand + Demon$) seems to provide an excellent performance in all tested instances. In addition, it is relatively easy to implement, since it does not require many parameters. Hence, we would recommend this one for most practical applications.

Three sets of classical benchmark instances have been adapted and extended in order to perform the computational experiments. The results for the deterministic version of the problem show the competitiveness of the proposed algorithm. In the case of stochastic demands, we have implemented Monte Carlo simulation to estimate the value of the safety stock strategy to be utilized as security buffer to face demand uncertainty. In addition, simulation has also been used to estimate stochastic costs and reliability levels for the obtained solutions.

In future research work, we plan to extend the simheuristic algorithm so it can also consider stochastic travel times and other corrective strategies (including the possibility of splitting the service). Additionally, it could be interesting to analyze a multi-objective version of the problem in which goals other than minimizing expected total costs can be targeted; e.g., minimizing variability of the stochastic solution.

Acknowledgements

This work has been partially supported by Rhenus Freight Logistics GmbH & Co. KG. We also acknowledge the support of the Erasmus+ Program (2019-I-ES01-KA103-062602).

References

Akca, Z., Berger, R., and Ralphs, T. (2009). A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In *Operations Research and Cyber-infrastructure*, pages 309–330. Springer.

Albareda-Sambola, M., Fernández, E., and Laporte, G. (2007). Heuristic and lower bound for a stochastic location-routing problem. *European Journal of Operational Research*, 179(3):940–955.

Baldacci, R., Mingozzi, A., and Calvo, R. W. (2011). An exact method for the capacitated location-routing problem. *Operations Research*, 59(5):1284–1296.

Barreto, S. (2004). *Análise e Modelização de Problemas de localização-distribuição*. PhD thesis, University of Aveiro, Portugal. Available at: https://ria.ua.pt/bitstream/10773/1850/1/2009001061.pdf, accessed: Dec-2018.

Barreto, S., Ferreira, C., Paixao, J., and Santos, B. S. (2007). Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3):968–977.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., and Calvo, R. W. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931–941.

Boudahri, F., Aggoune-Mtalaa, W., Bennekrouf, M., and Sari, Z. (2013). Application of a clustering based location-routing model to a real agri-food supply chain redesign. In *Advanced Methods for Computational Collective Intelligence*, pages 323–331. Springer.

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

Cordeau, J.-F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119.

Dai, Z., Aqlan, F., Gao, K., and Zhou, Y. (2019). A two-phase method for multi-echelon location-routing problems in supply chains. *Expert Systems with Applications*, 115:618–634.

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*, 68(10):1161–1176.

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

Duhamel, C., Lacomme, P., Prins, C., and Prodhon, C. (2010). A GRASPxELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37(11):1912–1923.

Efroymson, M. A. and Ray, T. L. (1966). A branch-bound algorithm for plant location. *Operations Research*, 14(3):361–368.

Escobar, J. W., Linfati, R., Baldoquin, M. G., and Toth, P. (2014). A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B: Methodological*, 67:344–356.

Faulin, J., Gilibert, M., Juan, A. A., Vilajosana, X., and Ruiz, R. (2008). SR-1: A simulation-based algorithm for the capacitated vehicle routing problem. In *Proceedings of the 2008 Winter Simulation Conference*, pages 2708–2716. IEEE.

Faulin, J. and Juan, A. A. (2008). The ALGACEA-1 method for the capacitated vehicle routing problem. *International Transactions in Operational Research*, 15(5):599–621.

Fikar, C., Juan, A. A., Martinez, E., and Hirsch, P. (2016). A discrete-event driven metaheuristic for dynamic home service routing with synchronised trip sharing. *European Journal of Industrial Engineering*, 10(3):323–340.

Fotakis, D. (2011). Online and incremental algorithms for facility location. *SIGACT News*, 42(1):97–131.

Fu, Y., Ding, J., Wang, H., and Wang, J. (2018). Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in industry 4.0-based manufacturing system. *Applied Soft Computing*, 68:847–855.

Fu, Y., Wang, H., Tian, G., Li, Z., and Hu, H. (2019). Two-agent stochastic flow shop deteriorating scheduling via a hybrid multi-objective evolutionary algorithm. *Journal of Intelligent Manufacturing*, 30(5):2257–2272.

Gendreau, M., Laporte, G., and Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing*, 27(12):1641–1653.

Ghaffari-Nasab, N., Jabalameli, M. S., Aryanezhad, M. B., and Makui, A. (2013). Modeling and solving the bi-objective capacitated location-routing problem with probabilistic travel times. *The International Journal of Advanced Manufacturing Technology*, 67(9-12):2007–2019.

Gillett, B. E. and Johnson, J. G. (1976). Multi-terminal vehicle-dispatch algorithm. *Omega*, 4(6):711–718.

Golden, B. L., Magnanti, T. L., and Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks*, 7(2):113–148.

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 flow-shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 79:23–36.

González-Neira, E. M., Montoya-Torres, J. R., and Caballero-Villalobos, J. P. (2019). A comparison of dispatching rules hybridised with monte carlo simulation in stochastic permutation flow shop problem. *Journal of Simulation*, 13(2):128–137.

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. (2016). 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. (2017). 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., Pérez, J. A. M., and Juan, A. A. (2018a). Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. *Computers & Industrial Engineering*, 123:278–288.

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

Guimarans, D., Dominguez, O., Panadero, J., and Juan, A. A. (2018). A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory*, 89:1–14.

Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228.

Henderson, D., Jacobson, S. H., and Johnson, A. W. (2003). The theory and practice of simulated annealing. In *Handbook of Metaheuristics*, pages 287–319. Springer.

Ho, W., Ho, G. T., Ji, P., and Lau, H. C. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4):548–557.

Hochbaum, D. S. (1982). Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on Computing*, 11(3):555–556.

Hof, J., Schneider, M., and Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an avns algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, 97:102–112.

Juan, A. A., Barrios, B. 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., Jorba, J., Caceres, J., and Marquès, J. M. (2013). 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. (2011). 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., Grasman, S. E., Caceres-Cruz, J., and Bektaş, T. (2014b). A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. *Simulation Modelling Practice and Theory*, 46:40–52.

Juan, A. A., Kelton, W. D., Currie, C. S., and Faulin, J. (2018). Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas. In *Proceedings of the 2018 Winter Simulation Conference*, pages 3048–3059. IEEE Press.

Juan, A. A., Pascual, I., Guimarans, D., and Barrios, B. (2015). Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem. *International Transactions in Operational Research*, 22(4):647–667.

Li, S. (2011). A 1.488 approximation algorithm for the uncapacitated facility location problem. In *International Colloquium on Automata, Languages, and Programming*, pages 77–88. Springer.

Lopes, R. B., Barreto, S., Ferreira, C., and Santos, B. S. (2008). A decision-support tool for a capacitated location-routing problem. *Decision Support Systems*, 46(1):366–375.

Lopes, R. B., Ferreira, C., and Santos, B. S. (2016). A simple and effective evolutionary algorithm for the capacitated location–routing problem. *Computers & Operations Research*, 70:155–162.

Marinakis, Y. (2015). An improved particle swarm optimization algorithm for the capacitated location routing problem and for the location routing problem with stochastic demands. *Applied Soft Computing*, 37:680–701.

Marinakis, Y., Marinaki, M., and Migdalas, A. (2016). A hybrid clonal selection algorithm for the location routing problem with stochastic demands. *Annals of Mathematics and Artificial Intelligence*, 76(1):121–142.

Mehrjerdi, Y. Z. and Nadizadeh, A. (2013). Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. *European Journal of Operational Research*, 229(1):75–84.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.

Mousavi, S. M. and Tavakkoli-Moghaddam, R. (2013). A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain. *Journal of Manufacturing Systems*, 32(2):335–347.

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

Nataraj, S., Ferone, D., Quintero-Araujo, C., Juan, A., and Festa, P. (2019). Consolidation centers in city logistics: A cooperative approach based on the location routing problem. *International Journal of Industrial Engineering Computations*, 10(3):393–404.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.

Prins, C., Prodhon, C., and Calvo, R. W. (2006a). A memetic algorithm with population management (ma— pm) for the capacitated location-routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 183–194. Springer.

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

Prins, C., Prodhon, C., Ruiz, A., Soriano, P., and Calvo, R. W. (2007). Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. *Transportation Science*, 41(4):470–483.

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

Quintero-Araujo, C. L., Caballero-Villalobos, J. P., Juan, A. A., and Montoya-Torres, J. R. (2017a). A biased-randomized metaheuristic for the capacitated location routing problem. *International Transactions in Operational Research*, 24(5):1079–1098.

Quintero-Araujo, C. L., Gruler, A., Juan, A. A., de Armas, J., and Ramalhinho, H. (2017b). Using simheuristics to promote horizontal collaboration in stochastic city logistics. *Progress in Artificial Intelligence*, 6(4):275–284.

Quintero-Araujo, C. L., Gruler, A., Juan, A. A., and Faulin, J. (2019). Using horizontal cooperation concepts in integrated routing and facility-location decisions. *International Transactions in Operational Research*, 26(2):551–576.

Renaud, J., Laporte, G., and Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23(3):229–235.

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

Schrage, L. (1978). Implicit representation of generalized variable upper bounds in linear programming. *Mathematical Programming*, 14(1):11–20.

Snyder, L. V. and Daskin, M. S. (2006). Stochastic p-robust location problems. *IIE Transactions*, 38(11):971–985.

Thangiah, S. R. and Salhi, S. (2001). Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem. *Applied Artificial Intelligence*, 15(4):361–383.

Thouin, F. and Coates, M. (2008). Equipment allocation in video-on-demand network deployments. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 5(1):5.

Tillman, F. A. and Cain, T. M. (1972). An upperbound algorithm for the single and multiple terminal delivery problem. *Management Science*, 18(11):664–682.

Ting, C.-J. and Chen, C.-H. (2013). A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics*, 141(1):34–44.

Yu, V. F., Lin, S.-W., Lee, W., and Ting, C.-J. (2010). A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering*, 58(2):288–299. Scheduling in Healthcare and Industrial Systems.

Zarandi, M. H. F., Hemmati, A., Davari, S., and Turksen, I. B. (2013). Capacitated location-routing problem with time windows under uncertainty. *Knowledge-Based Systems*, 37:480–489.

Zhang, S., Chen, M., and Zhang, W. (2019). A novel location-routing problem in electric vehicle transportation with stochastic demands. *Journal of Cleaner Production*, 221:567–581.