HDR

Jorge E. Mendoza

May 15, 2017

Contents

Chapter 1

Introduction

Chapter 2

Technical background

In this chapter we briefly overview the main optimization techniques used on our research. We assume the reader is familiar with all these techniques, but for the sake of completeness we provide him/her with just enough technical background to easily navigate the rest of the document. We begin by introducing local search-based algorithms and quickly survey their application to vehicle routing problems. We then discuss evolutionary algorithms and highlight successful applications to VRPs. Next, we focus on order-first, split-second techniques. Finally we discuss the multi-space sampling heuristic; a tecnique

2.1 Order-first, split-second procedures

2.2 Route-first, Assemble-second heuristics

2.3 Local search-based metaheuristics

2.4 Evolutionary algorithms

Chapter 3

Stochastic and dynamic routing

Stochastic vehicle routing problems (SVRPs) were introduced in the late 60's by Tillman (1969). In a nutshell SVRPs consider than one or more problem parameters (e.g., demands, travel times) are unknown when the routes are planned. Despite being around for nearly 50 years, SVRPs have attracted much less attention than their deterministic counterparts. This is a somehow intriguing phenomenon considering that in practice most VRPs are stochastic by nature. This status quo is, however, slowly changing. With recent advances on big data and intelligent transportation systems, the industry is more demanding on routing technology that can exploit the knowledge encapsulated in the massively available amounts of data. The routing community has responded positively to this challenge. As Gendreau et al. (2016) point out, in the last 15 years the effort (and therefore the number of publications) devoted to SVRPs has substantially increased. This chapter summarizes our contributions to that effort.

3.1 Stochastic demands

The vehicle routing problem with stochastic demands (VRPSD) is without any doubts the most studied variant of SVRP. In the VRPSD a set of geographically spread customers demand (or supply) a product that must be delivered (or collected) using a fleet of limited-capacity vehicles located at a central depot. The particular characteristic of the problem is that the exact quantities demanded (supplied) by each customer are only known upon the vehicle's arrival at the customer location (i.e., they are stochastic). It is assumed, however, that each customer's demand follows a known probability distribution. The main impact of stochastic demands is that they introduce uncertainty into the feasibility of the routes; depending on the demand realizations (i.e., the actual values), a vehicle may arrive at a customer without enough capacity to satisfy its demand.

To deal with uncertain demands in the VRPSD, researchers have explored models based on various solution frameworks including chance-constraint programming, stochastic programming with recourse, dynamic programming, Markov decision models, and the multi-scenario approach. Each of these frameworks takes into accounts factors such as instance size, assumptions about available technology (e.g., real-time communication between vehicles and decision-makers), and assumptions about managerial policies (e.g., whether or not routes can be modified during their execution). For a complete discussion of the characteristics of each framework the reader is referred to Secomandi and Margot (2009) and Pillac et al. (2013).

The most widely studied models in the literature are those based on two-stage stochastic programming (Cordeau et al., 2006). As the name suggests, in this framework the problem is solved in two phases. In the first phase a set of *a priori routes* is planned, and in the second phase the routes are executed. If there is a capacity constraint violation, or *route failure*, a corrective action, known as *recourse*, is taken to recover feasibility. In general, the recourse actions generate an extra cost known only after the second phase. Thus, the objective is to design during the first phase a set of routes that minimizes the sum of the cost of the a priori routes and the expected cost of the recourse actions.

The most traditional recourse action, known as *detour-to-depot*, involves traveling back to the depot to restore the vehicle capacity, returning to the customer to complete the service, and then continuing the route as initially planned (Savelsbergh and Goetschalckx, 1995). However, more sophisticated approaches have been explored in the literature. These include performing preemptive trips to the depot in an attempt to avoid route failures (Yang et al., 2000; Bianchi et al., 2004; Tatarakis and Minis, 2009; Pandelis et al., 2012), assigning each vehicle a partner to provide back-up in the event of a failure (Ak and Erera, 2007), and assigning customers to two routes and move them from their primary to their backup route in case of a failure (Erera et al., 2009). Although the detour-to-depot has been criticized for its apparent “lack of realism”, this policy provides a simple way to deal with route failures in practice. Moreover, Gendreau et al. (2016) also point out that the detour-to-depot policy produces routes that are *tactically stable* (i.e., they do not vary much from their initial plan); a desirable characteristic from an operational perspective. From a more academic perspective, we would add that given the massive literature dealing with models built on top of it, the detour-to-depot policy provides a solid base to assess new algorithms for the VRPSD. All the contributions described in this subsection relay on models employing the detour-to-depot policy.

Formally, the VRPSD can be defined on a complete and undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{0, \dots, n\}$ is the vertex set and $\mathcal{E} = \{(v, u) : v, u \in \mathcal{V}, v \neq u\}$ is the edge set. Vertices $v = 1, \dots, n$ represent the customers and vertex $v = 0$ represents the depot. A weight t_e is associated with edge $e = (v, u) = (u, v) \in \mathcal{E}$, and it represents the travel time between vertices v and u. Each customer v has a random demand $\tilde{\xi}_v$ for a given product. The customers are served using an unlimited fleet of homogeneous vehicles with capacity Q. In general, it is assumed that: i) each customer’s demand follows an independent and known probability distribution, ii) the demand realizations $\vec{\xi}$ are nonnegative and less than the capacity of the vehicle, and iii) each customer’s demand realization is not known until the vehicle arrives at the customer location.

A planned route r is a sequence of vertices $r = (0, v_1, \dots, v_i, \dots, v_{n_r}, 0)$, where $v_i \in \mathcal{V} \setminus \{0\}$ and n_r is the number of customers served by the route. Depending on the context, we may refer to route r as an ordered set of edges $r = \{(0, v_1), \dots, (v_{i-1}, v_i), \dots, (v_{n_r}, 0)\}$. During the execution of a planned route, if a route failure occurs, that is, the capacity of the vehicle is exceeded, the detour-to-depot recourse is applied to recover the feasibility of the route. We denote by $Pr(v_i)$ the probability of a route failure occurring while serving customer $v_i \in r$. This *failure probability* is given by

$$Pr(v_i) \quad = \quad \sum_{i=2}^{n_r} \sum_{f=1}^{i-1} Pr\left(\sum_{j=2}^{i-1} \tilde{\xi}_{v_j} \leq f \cdot Q < \sum_{j=2}^{i} \tilde{\xi}_{v_j} \right) \tag{3.1}$$

where the probability term represents the probability of the f^{th} failure occurring while serving customer v_i. For the details of the derivation of (3.1) see (Teodorović and Pavković, 1992). Note that the number and location of route failures are not known when the routes are planned. Therefore, although all travel times are (assumed to be) deterministic, the total duration of a route $\tilde{T}_r$ is a random variable which realization is only known when the route is completed. On the other hand, the probability distribution of $\tilde{T}_r$ may be computed when the route is planned (see § 3.1.2 for further details).

The VRPSD consists in determining a set $\mathcal{R}$ of planned routes that minimizes:

$$E\left[C\right] = \sum_{r \in \mathcal{R}} E\left[l_r + G_r\left(\vec{\xi}\right)\right] = \sum_{r \in \mathcal{R}} l_r + \sum_{r \in \mathcal{R}} E\left[G_r\left(\vec{\xi}\right)\right] \tag{3.2}$$

where l_r denotes the planned length (planned cost) and $E\left[G_r\left(\vec{\xi}\right)\right]$ the expected length of the returning trips to the depot, or cost of recourse, caused by route failures for each route $r \in \mathcal{R}$. The planned cost of a route is given by the sum of the lengths of the arcs traversed by the route. On the other hand, the estimation of the expected cost of recourse is slightly more complicated. Under the detour-to-depot recourse action, the expected cost of the failures in a route is given by:

$$E\left[G_r\left(\vec{\xi}\right)\right] \quad = \quad 2 \times \sum_{i=2}^{n_r} \sum_{l=1}^{i-1} Pr\left(\sum_{j=2}^{i-1} \xi_{v_j} \leq l \cdot Q < \sum_{j=2}^{i} \xi_{v_j}\right) \times d_{v_i,0} \tag{3.3}$$

The expected cost of failures in (3.3) can be efficiently computed when customer demands follow a probability function with the cumulative property. This property states that the sum of two independent and Ψ distributed random variables is also Ψ distributed, as it is the case for the normal, Poisson, and Gamma distributions.

3.1.1 Solving the VRPSD

The literature reports on several exact and heuristic approaches to tackle the VRPSD. Exact methods include that of Laporte et al. (2002) who proposed an implementation of the L-Shaped algorithm and solved to optimality instances of up to 50 and 100 customers of a variant with limited fleet. In the same vein, Rei et al. (2007) proposed an implementation of the L-Shaped algorithm with local branching cuts for a variant in which a single route servicing all customers is to be designed (this problem is usually referred in the literature as the SVRPSD). The authors reported optimal solutions for instances of up to 90 customers with uniformly distributed demands. Christiansen and Lysgaard (2007) proposed a branch-and-price algorithm to tackle the baseline formulation. Their approach successfully solved instances of up to 60 customers with Poisson distributed demands. That algorithm was recently re-implemented and improved by Gauvin et al. (2014) who currently hold the best exact results for the problem. In the segment of heuristic approaches, Gendreau et al. (1996) proposed a tabu search (TS) algorithm, known as *tabustoch*, designed to tackle an extension of the baseline formulation in which, in addition to the demands, customers are also stochastic (i.e., they are present, or not, with a given probability). Yang et al. (2000) introduced two constructive heuristics for another extended formulation in which preventive trips to the depot are allowed. A similar formulation was proposed by Bianchi et al. (2006) in the context of the SVRPSD. To solve their problem, these authors introduced a set of metaheuristics comprising simulated annealing (SA), iterated local search, ant colony optimization, evolutionary algorithms, and TS. More recently, Rei et al. (2010) introduced a hybrid Monte Carlo local branching approach for the SVRPSD. Goodson et al. (2012) introduced a hybrid simulated annealing that embeds sophisticated neighborhood schemes into a local search procedure. To our knowledge the most recent metaheuristic for the VRPSD is the variable neighborhood search algorithm of Sarasola et al. (2016).

In joint work with Juan G. Villegas (UdeA, Colombia) we proposed the multi-space sampling heuristic (MSH) to solve the VRPSD. MSH works in two phases. In the first phase (sampling phase), it uses a set H of *sampling heuristics* to draw T elements from the set of TSP-like tours (i.e., $\mathcal{P}$) and maps each sampled element p^t to a sub-set Ω^t in the set of feasible routes (i.e., $\mathcal{T}$). In the second phase (mapping phase), the approach maps a set $\Omega \subset \mathcal{T}$, where $\Omega = \Omega^1 \cup \ldots \cup \Omega^T$, to one element $s \in \mathcal{S}$ by solving a set-partitioning formulation of the problem. Algorithm 1 presents the general structure of the proposed heuristic.

To draw elements from $\mathcal{P}$ (line 7 in Algorithm 1), we use randomized versions of 4 different TSP constructive heuristics: randomized nearest neighbor (RNN), randomized nearest insertion (RNI), randomized best insertion (RBI), and randomized farthest insertion (RFI). Although the strategies used to generate the randomized versions of the four heuristics are rather intuitive, for the sake of completeness we briefly describe them here.

Let p be the TSP tour being built by a given sampling heuristic, $\mathcal{W}$ the set of vertices visited by p, and $\mathcal{N} = \mathcal{V} \setminus \mathcal{W}$ an ordered set of not-routed vertices. For the sake of simplicity, we assume that sets $\mathcal{W}$ and $\mathcal{N}$ are updated every time a customer is added to p. Let us also define three metrics for every customer $v \in \mathcal{N}$, namely, $d_{min}(v) = \min\{d_{v,u} | u \in \mathcal{W}\}$, $d_{max}(v) = \max\{d_{v,u} | u \in \mathcal{W}\}$, and $\Delta_{min}(v) = \min\{d_{u,v} +$

Algorithm 1 Multi-space sampling heuristic: general structure

```
 1: function MultiSpaceSampling($G$,$H$)
 2:     $t \leftarrow 1$
 3:     $\Omega \leftarrow \emptyset$
 4:     while $t \leq T$ do
 5:         for $k = 1$ to $k = |H|$ do
 6:             $h \leftarrow H_k$
 7:             $p^t \leftarrow h(G)$
 8:             $\langle \Omega^t, s^t \rangle \leftarrow s\text{-}split(G, p^t)$
 9:             $\Omega \leftarrow \Omega \cup \Omega^t$
10:             if $t = 1$ then
11:                 $s^* \leftarrow s^t$
12:             else if $f(s^t) < f(s^*)$ then
13:                 $s^* \leftarrow s^t$
14:             end if
15:             $t \leftarrow t + 1$
16:         end for
17:     end while
18:     $s^* \leftarrow setPartitioningProblem(G, \Omega, s^*)$
19:     return $\mathcal{R} \leftarrow s^*$
20: end function
```

$d_{v,w} - d_{u,w}|(u,w) \in p\}$. Finally, let k be a random integer selected in $[1, \min\{K, |\mathcal{N}|\}]$, where parameter K denotes the *randomization factor* of each heuristic. The four sampling heuristics operate as follows:

- **RNN:** Set $p = \{0\}$ and $u = 0$. At each iteration: identify the vertex v who is the k_{th} nearest vertex to u, append v to p, and set $u = v$. Stop when $|\mathcal{N}| = 0$ and append 0 to p to complete a tour.
- **RNI:** Initialize p as a tour starting at the depot and performing a round trip to a randomly selected customer (henceforth this procedure will be referred simply as initialize p). At each iteration: sort $\mathcal{N}$ in non-decreasing order of $d_{min}(v)$. Insert $v = \mathcal{N}_k$ in the best possible position in the tour (i.e., the position generating the shortest increment in the cost of the tour). Stop when $|\mathcal{N}| = 0$.
- **RFI:** Initialize p. At each iteration: sort $\mathcal{N}$ in non-decreasing order of $d_{max}(v)$ and insert $v = \mathcal{N}_k$ in the best possible position in the tour. Stop when $|\mathcal{N}| = 0$.
- **RBI:** Initialize p. At each iteration: sort $\mathcal{N}$ in non-decreasing order of $\Delta_{min}(v)$ and insert $v = \mathcal{N}_k$ in the best possible position in the tour. Stop when $|\mathcal{N}| = 0$.

To map each sampled element p^t to its corresponding $\Omega^p \subset \mathcal{T}$ (line 8 in Algorithm 1) our approach uses the s-split procedure for the VRPSD Mendoza et al. (2010). S-split was originally proposed to map an element in $\mathcal{P}$ to an element in $\mathcal{S}$ by optimally partitioning the giant tour into a set of feasible routes that form a VRPSD solution. Nonetheless, s-split accomplishes its mission by running a dynamic programming algorithm that evaluates every single feasible route that can be obtained from the giant tour without altering the order of the customers. In other words, by saving a reference to the set of routes evaluated by s-split while partitioning p^t, we obtain a mapping of p^t to a sub-set $\Omega^t \subset \mathcal{T}$. Since s-split also retrieves a mapping of p^t to a solution $s^t \in \mathcal{S}$, we keep a reference to the best solution found during the sampling phase (lines 10–14 in Algorithm 1) to have a good upper bound in the second phase of our heuristic. The leftmost frame in Figure 3.1 illustrates the operation of s-split in an execution of the proposed heuristic with $H = \{RFI, RNI\}$ and $T = 2$. For all details needed to implement s-split the reader is referred to Mendoza et al. (2010, 2011).

In the second phase, our approach maps the set $\Omega = \Omega^1 \cup \ldots \cup \Omega^T$ to a solution $s^* \in \mathcal{S}$ by solving a set-partitioning formulation of the VRPSD proposed by Christiansen and Lysgaard (2007) $\big(\min_{\mathcal{R} \subseteq \Omega} \big\{\sum_{r \in \mathcal{R}} E[C_r] : \bigcup_{r \in \mathcal{R}} = \mathcal{V}$ The objective is then to select the best sub-set of routes from Ω to build the set of planed routes $\mathcal{R}$ (i.e., solution) ensuring that each customer will be visited by exactly one route. The rightmost frames in Figure 3.1 illustrate the operation of the mapping phase in our heuristic.

In most VRP variants, it is possible to replace the set-partitioning formulation of the problem by an equivalent set-covering formulation aiming to gain in computational efficiency. This approach is based in the premise that the cheapest way to cover all customers is to cover them only once because the cost verifies the triangle inequality. Therefore, the optimal solution of the set-covering problem is a feasible VRP solution. In the VRPSD, however, that is not necessarily the case. Indeed, because the expected cost of failing while visiting a customer v depends on the expected demands of all customers previously visited by the route and on its distance to the depot, see Equation (**??**), there is no guarantee that a route $r = (0, \ldots, u, v, \ldots, 0)$ has a total expected cost $E[C_r]$ that is cheaper than that of a route $r' = (0, \ldots, u, w, v, \ldots, 0)$. Hence, the cheapest way to cover all customers may include covering some customers more than once, which leads to an unfeasible VRPSD solution. One can overcome this difficulty by implementing an ad-hoc verification and reparation procedure. After some preliminary experimentation using a set-covering formulation in the mapping phase, we assessed that the benefits obtained in terms of computational efficiency did not pay off the loss in simplicity of our heuristic, so we kept the original set-partitioning formulation.

It is worth mentioning that although in many VRP variants excellent solutions can be found by solving set-covering/set-partitioning formulations over a reduced set of routes, researchers often oversight the benefits of embedding this kind of component in heuristic procedures. Authors like Villegas et al. **?**, however, have shown that applied as a post-optimization procedure, this approach may lead to significant improvements in the quality of final solutions. We consider important to remark that in the proposed heuristic, the set-partitioning model is an integral part of the method rather than only a post-optimization strategy.

3.1.2 Handling duration constraints

One of the most challenging issues when solving the VRPSD and its variants is dealing with duration constraints (DCs). It is easy to see that recourse actions add travel time to the planned routes. Since the exact number of recourses and the extra time they add to each route is not known when the routes are planned, the total duration of a route is itself a random variable. The latter may lead to a problem in practice if the routes are subject to DCs.

Duration constraints have been studied only rarely in the context of the VRPSD. To the best of our knowledge, the body of work in this domain is limited to about ten references, most of them focusing on approaches based on two-stage stochastic programming. For the sake of brevity, in the remainder of this subsection we focus on these approaches; however, we refer the reader to the excellent papers by Bent and Van Hentenryck (2004, 2007) and Goodson et al. (2013a,b) for research based on other frameworks.

Yang et al. (2000) is probably the first reference to DCs in the VRPSD literature. The authors handle these constraints by imposing a limit on the expected duration of the a priori routes. Mendoza et al. (2010, 2011) applied the same strategy in the context of the multi-compartment VRPSD (MC-VRPSD), a problem in which each customer demands several incompatible products that are transported in different vehicle compartments. The main advantage of this *constrained expected duration* approach is its computational convenience. Indeed, since the expected duration of a route is usually computed as part of the objective function, the DC feasibility check requires no additional effort. On the other hand, although this strategy may be adequate for practical situations where DCs are rather soft constraints, it does not provide decision-makers with an explicit mechanism to express their preferences about violations of these constraints.

Tan et al. (2007) and Sörensen and Sevaux (2009) propose an alternative approach, based on penalizing violations of the DCs in the objective function. Tan et al. (2007) use the penalties as part of a cost function called drivers' remuneration that they optimize, along with the total traveled distance and the number of vehicles, using a multi-objective optimization approach. Sörensen and Sevaux (2009) include the penalties directly in the total-duration objective function and use an established mono-objective approach (Sörensen and Sevaux, 2006) to solve the problem. In both cases, the authors use Monte Carlo simulation to generate multiple scenarios of the demand realizations that are used to compute the total expected duration of the routes and the penalties for DC violations. Mendoza et al. (2009) propose a different strategy to address DCs in the context of a bi-objective MC-VRPSD: they minimize simultaneously the total expected cost of a set of routes and its coefficient of variation. In their approach, DCs are imposed on planned routes as *chance constraints* ensuring that the probability of completing a route in less than its maximum duration is greater than a given threshold. To perform the feasibility check of the chance constraints, the authors use Monte Carlo simulation.

From the conceptual point of view, both the penalty and chance-constraint approaches overcome the short-

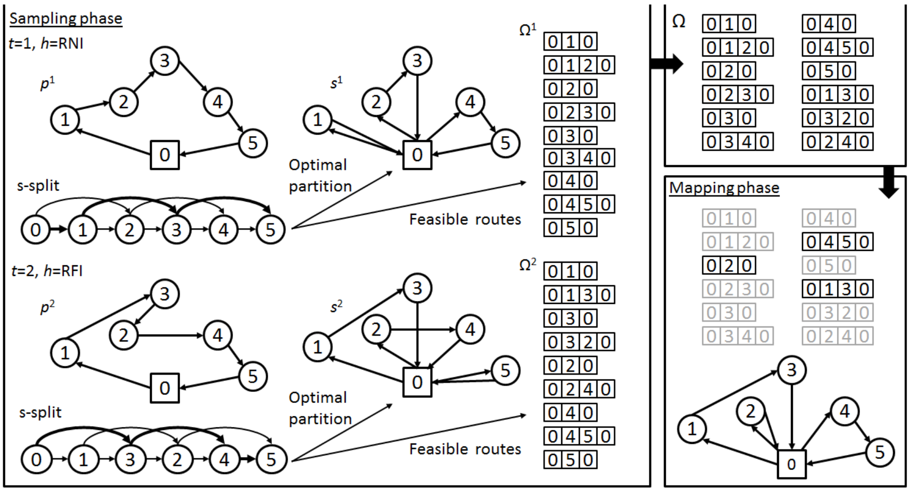


Figure 3.1: Multi-space sampling heuristic: example of an execution with $H = \{RNI, RFI\}$ and $T = 2$

comings of the constrained expected-duration approach. However, the implementations based on Monte Carlo simulation may be unnecessarily expensive from a computational point of view because one may need to generate a large number of scenarios to achieve statistical significance. Haugland et al. (2007) and Erera et al. (2010) propose approaches for applications in which the DCs are hard constraints. In Haugland et al. (2007) the authors solve a VRPSD with DCs as part of the evaluation of the solution to a districting problem. To check the DC feasibility the authors use an upper bound on the total duration of a route. Erera et al. (2010) propose an algorithm to compute the maximum duration of a route for any realization of the customer demands. They use its result as an input to check the DCs.

In joint work with J.G. Villegas (UdeA, Colombia) and L.-M. Rousseau (CIRRELT, Canada), carried between 2012 and 2013, we revisited the penalty and chance-constraint strategies to deal with DCs in the VRPSD. One of the most important contributions of our work is that in contrast to previous approaches, we did not use Monte Carlo simulation to evaluate routes. We instead devised a mechanism for explicitly building the probability distribution of the duration of a route. In the remainder of this subsection we briefly discuss this and other contributions to the VRPSD with DCs (VRPSD-DC); full details can be found in (Mendoza et al., 2016).

Computing the probability distribution of the duration of a route

Note that according to the VRPSD definition given in **??** all the demand realizations are less than the capacity of the vehicle. Therefore, a route may fail at the most once at each customer but the first one. In other words, the maximum number of failures in a route is $n_r - 1$, and the first failure cannot occur while serving the first customer. Consequently, $\tilde{T}_r$ follows a discrete distribution with 2^{n_r-1} possible outcomes; we refer to each of these outcomes as a *duration profile*. Figure 3.2 illustrates this concept. Let $\mathcal{P}(r)$ be the set of all possible duration profiles for route r. Let $Pr(p)$ be the probability of observing duration profile $p \in \mathcal{P}(r)$, and let $T_r(p)$ be the total duration of route r if profile p is observed. The leftmost part of the figure depicts a planned route $r = \{(0,a),(a,b),(b,c),(c,0)\}$ with a total planned duration $t_r = \sum_{(u,v)\in r} t_{(u,v)} = t_{(0,a)} + t_{(a,b)} + t_{(b,c)} + t_{(c,0)}$. The rightmost part of the figure shows a tree in which each leaf node represents one possible duration profile p for route r. For instance, leaf node $p = 1$ represents the duration profile of the route if failures occur while serving customers b and c. In such case, $T_r(p) = t_r + 2t_{(b,0)} + 2t_{(c,0)}$ and $Pr(p) = (1 - Pr(a)) \times Pr(b) \times Pr(c)$, where $Pr(a)$, $Pr(b)$, and $Pr(c)$ denote the probability of having a failure while servicing customers a, b, and c, respectively (see Equation (3.1)). It is worth mentioning that since failures cannot occur while servicing customer a, the upper branches of the three (in grey) represent impossible duration outcomes. Building on top of our duration profiles we proposed two stochastic programming formulation for the VRPSD-DC.

Chance-constraint formulation

In our first formulation we extend the classical two-stage stochastic programming formulation for the VRPSD to include the DCs as chance constraints. The resulting problem involves finding a set $\mathcal{R}$ of planned routes that minimizes

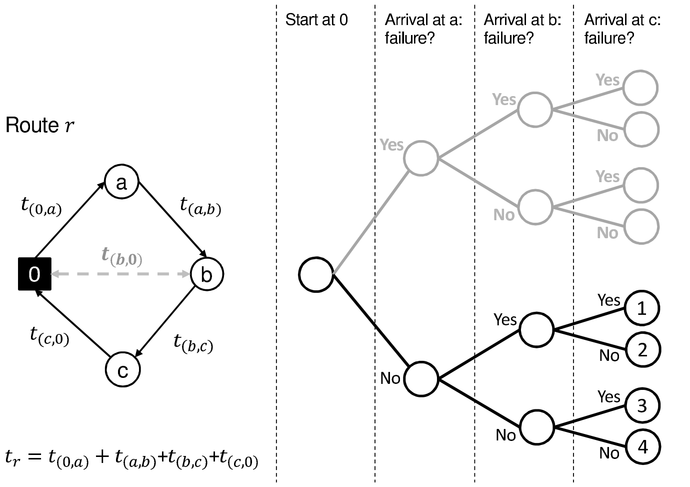
$$E\left[C_1(\mathcal{R})\right] = \sum_{r\in\mathcal{R}} E\left[\tilde{T}_r\right] \tag{3.4}$$

s.t.

$$\sum_{r\in\mathcal{R}} Pr\left(\tilde{T}_r > T\right) \leq \beta \qquad \forall\, r \in \mathcal{R} \tag{3.5}$$

$$\sum_{i\in r} E\left[\tilde{\xi}_{v_i}\right] \leq Q \qquad \forall\, r \in \mathcal{R} \tag{3.6}$$

$$r \bigcap r' = \{0\} \qquad \forall\, r, r' \in \mathcal{R}, r \neq r' \tag{3.7}$$



p	$T_r(p)$	$Pr(p)$
1	$t_r + 2t_{(b,0)} + 2t_{(c,0)}$	$(1 - Pr(a)) \times (Pr(b)) \times (Pr(c))$
2	$t_r + 2t_{(b,0)}$	$(1 - Pr(a)) \times (Pr(b)) \times (1 - Pr(c))$
3	$t_r + 2t_{(c,0)}$	$(1 - Pr(a)) \times (1 - Pr(b)) \times (Pr(c))$
4	t_r	$(1 - Pr(a)) \times (1 - Pr(b)) \times (1 - Pr(c))$

Figure 3.2: Duration profiles for a given route and their attributes

$$\bigcup_{r\in\mathcal{R}} = \mathcal{V} \tag{3.8}$$

The objective (3.4) minimizes the total expected duration of the set of routes $\mathcal{R}$. Constraint (3.5) ensures that the probability that a route violates the duration limit is lower than a given threshold β. Using the duration profiles of route r as an input, the first term in (3.5) can be computed as

$$Pr\left(\tilde{T}_r > T\right) = \sum_{p\in\mathcal{P}(r)|T_r(p)>T} Pr(p). \tag{3.9}$$

Constraint (3.6) ensures that each planned route is designed so that the total expected load does not exceed the capacity of the vehicle while constraints (3.7) and (3.8) guarantee that each customer is included in one and only one planned route.

Penalty formulation

In our second formulation we follow a completely different approach. To account for the DCs, we extend the classical VRPSD objective to include the expected cost of overtime, i.e., the time that each route travels above the limit T. In this formulation the problem involves finding a set of planned routes $\mathcal{R}$ verifying constraints (3.6)–(3.8) and minimizing

$$E\left[C_2(\mathcal{R})\right] = \sum_{r\in\mathcal{R}} E\left[\tilde{T}_r\right] + E\left[\phi\left(\tilde{O}_r\right)\right] \tag{3.10}$$

where

$$E\left[\phi\left(\tilde{O}_r\right)\right] = \sum_{p\in\mathcal{P}(r)|T_r(p)>T} \phi\left(T_r(p) - T\right) \times Pr(p) \tag{3.11}$$

is the expected overtime cost. Function $\phi(\cdot)$ models the decision maker's aversion toward overtime. It can take any form depending on the context. For instance, quadratic functions can model situations in which even small violations of the DCs should be discouraged, while piecewise linear functions can be useful when small violations are acceptable but violations beyond a given threshold should be avoided.

In the remainder of the subsection, we refer to our chance-constraint and penalty formulations as `CC` and `PF`.

A hybrid metaheuristic

To solve our two formulations for the VRPSD-DC, namely `CC` and `PF`, we developed a GRASP with HC. Algorithm 2 describes the proposed approach. At the kth GRASP iteration (lines 3–14) we greedily construct a starting solution (lines 5–6) and then try to improve it using a local search procedure (line 7). To construct the starting solution, we select a randomized TSP heuristic h from a predefined set $\mathcal{H}$ and use it to build a giant TSP tour tsp^k visiting all the customers (line 5). We then use an adaptation of the s-split procedure for the VRPSD (Mendoza et al., 2011) to optimally partition tsp^k into a set of feasible routes that forms a starting solution s^k (line 6). We next launch a VND procedure from the starting solution s^k (line 7). At the end of iteration k, we update the best solution s^* (line 8) and add the routes of the local optimum (i.e., s^k) to a set Ω (lines 9–11). After K iterations the GRASP stops and we carry out the HC. In this phase, our method solves a set partitioning problem (SPP) over the set of routes Ω (line 15). Note that the specific implementations of `split`$(\cdot)$ and `vnd`$(\cdot)$ vary depending on the formulation (i.e., `CC` or `PF`) being solved, whereas the implementations of `tsp`$(\cdot)$, `update`$(\cdot)$, and `setPartitioning`$(\cdot)$ are unchanged. For a thorough description of the algorithmic components embedded into our GRASP as well as a full description of how the split procedure was adapted to work on `CC` and `PF` the reader is referred to Mendoza et al. (2016).

Algorithm 2 GRASP+HC: General structure

```
function GRASPHC($\mathcal{H}$,$K$,mode)                    ▷ mode={CC, PF}
    $\Omega \leftarrow \emptyset$, $k \leftarrow 1$
    while $k \leq K$ do
        for $h \in \mathcal{H}$ do
            $tps^k \leftarrow$tsp($h$)
            $s^k \leftarrow$split($tsp^k$, mode)
            $s^k \leftarrow$vnd($s^k$, mode)
            $s^* \leftarrow$update($s^k$, $s^*$)
            for $r \in s^k$ do
                $\Omega \leftarrow \Omega \cup r$
            end for
            $k \leftarrow k + 1$
        end for
    end while
    $\mathcal{R} \leftarrow$ setPartitioning($\Omega, s^*$)
    return $\mathcal{R}$
end function
```

Some results

Classical VRPSD
It is worth nothing that solving the classical VRPSD is equivalent to solving CC with $\beta = 1$ (i.e., the DC becomes redundant). For validation purposes, we tested our approach on the classical VRPSD. We ran our hybrid metaheuristic on the 40-instance testbed of Christiansen and Lysgaard (2007). These instances range from 16 to 60 customers and assume Poisson-distributed demands. To assess the effectiveness of our method, we compared our results to the best known solutions (BKSs) for the testbed: 38 optimal solutions reported by Gauvin (2012) and 2 heuristic solutions reported by Goodson et al. (2012) and Mendoza and Villegas (2013).

The results showed that in terms of solution quality our approach outperforms the two state-of-the-art metaheuristics: it matched the 40 BKSs for the set, whereas Mendoza and Villegas (2013) achieved 27/40 and Goodson et al. (2012) achieved 33/40. Although it was difficult to make a precise comparison of the computational performance because of slight differences in the testing environments, the results also suggested that our approach outperforms the two other methods on this measure. To the best of our knowledge, our approach is still the state-of-the-art metaheuristic to tackle the VRPSD, despite the more recent publication of the Sarasola et al. variable neighborhood search.

3.1.3 The case of multi-compartment vehicles

Papers: Mendoza et al. (2010,2011)

3.1.4 The trade-off between expected cost and variance

Conference paper: Mendoza et al. (2009)

3.2 Stochastic travel and service times

Paper: Gmez et al. (2016)

3.2.1 Handling correlated parameters

Working paper: Sarmiento et al. (2016)

3.3 Perspectives

3.3.1 Combining optimization and machine learning

On going Ph.D. Thesis: Mustapha Haouassi

Bibliography

A. Ak and A. Erera. A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. *Transportation Science*, 41(2):222–237, 2007.

R. Bent and P. Van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.

R. Bent and P. Van Hentenryck. Waiting and relocation strategies in online stochastic vehicle routing. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1816–1821, 2007.

L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto. Metaheuristics for the vehicle routing problem with stochastic demands. In *Parallel Problem Solving from Nature - PPSN VIII*, Lecture Notes in Computer Science, pages 450–460. Springer Berlin, Heidelberg., 2004.

L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms*, 5(1):91–110, 2006.

C. Christiansen and J. Lysgaard. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters*, 35(6):773–781, 2007.

J. F. Cordeau, G. Laporte, M. Savelsbergh, and D. Vigo. Vehicle routing. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science: Transportation*, volume 14, pages 367–428. Elsevier, Amsterdam, 2006.

A. Erera, M. Savelsbergh, and E. Uyar. Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. *Networks*, 54(4):270–283, 2009.

A. Erera, J. C. Morales, and M. Savelsbergh. The vehicle routing problem with stochastic demands and duration constraints. *Transportation Science*, 44(4):474–492, 2010.

C. Gauvin. Un algorithme de génération de colonnes pour le problème de tournées de véhicules avec demandes stochastiques. Master's thesis, École Polytechnique de Montréal, 2012.

C. Gauvin, G. Desaulniers, and M. Gendreau. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, 50(0):141 – 153, 2014. ISSN 0305-0548.

M. Gendreau, G. Laporte, and R. Séguin. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research*, 44(3):469–477, 1996.

M. Gendreau, O. Jabali, and W. Rei. 50th anniversary invited articlefuture research directions in stochastic vehicle routing. *Transportation Science*, 50(4):1163–1173, 2016. doi: 10.1287/trsc.2016.0709.

J. C. Goodson, J. W. Ohlmann, and B. W. Thomas. Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. *European Journal of Operational Research*, 227(2):312–323, 2012. ISSN 0377-2217.

J. C. Goodson, J. W. Ohlmann, and B. W. Thomas. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. Technical report, University of Iowa, 2013a.

J. C. Goodson, J. W. Ohlmann, and B. W. Thomas. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. *Operations Research*, 61(1):138–154, 2013b. ISSN 0030-364X.

D. Haugland, S. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.

G. Laporte, F. Louveaux, and L. Van Hamme. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

J. E. Mendoza and J. G. Villegas. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optimization Letters*, 7(7):1503–1516, 2013.

J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco. A simulation-based MOEA for the multi-compartment vehicle routing problem with stochastic demands. In *Proceedings of the 8th metaheuristics international conference (MIC). Hamburg, Germany*, 2009.

J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37(11):1886–1898, 2010.

J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco. Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transportation Science*, 45(3):335–345, 2011.

J. E. Mendoza, L. M. Rousseau, and J. G. Villegas. A hybrid metaheuristic for the vehicle routing problem with stochastic demands and duration constraints. *Journal of Heuristics*, 22(4):539–566, 2016.

D. Pandelis, E. Kyriakidis, and T. Dimitrakos. Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. *European Journal of Operational Research*, 217(2):324 – 332, 2012. ISSN 0377-2217. doi: http://dx.doi.org/10.1016/j.ejor.2011.09.027. URL //www.sciencedirect.com/science/article/pii/S0377221711008551.

V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013. ISSN 0377-2217.

W. Rei, M. Gendreau, and P. Soriano. Local branching cuts for the 0-1 integer L-shaped algorithm. (CIRRELT-2007-23), 2007.

W. Rei, M. Gendreau, and P. Soriano. A hybrid Monte Carlo local branching algorithm for the single vehicle routing problem with stochastic demands. *Transportation Science*, 44(1):136–146, 2010.

B. Sarasola, K. F. Doerner, V. Schmid, and E. Alba. Variable neighborhood search for the stochastic and dynamic vehicle routing problem. *Annals of Operations Research*, 236(2):425–461, 2016. ISSN 1572-9338. doi: 10.1007/s10479-015-1949-7. URL http://dx.doi.org/10.1007/s10479-015-1949-7.

M. Savelsbergh and M. Goetschalckx. A comparison of the efficiency of fixed versus variable vehicle routes. *Journal of Business Logistics*, 16:163–187, 1995.

N. Secomandi and F. Margot. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, 57(1):214–230, 2009.

K. Sörensen and M. Sevaux. MA|PM: memetic algorithms with population management. *Computers & Operations Research*, 33(5):1214–1225, 2006.

K. Sörensen and M. Sevaux. A practical approach for robust and flexible vehicle routing using metaheuristics and Monte Carlo sampling. *Journal of Mathematical Modelling and Algorithms*, 8(4):387–407, 2009.

K. C. Tan, C. Y. Cheongas, and C. K. Goh. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, 177(2):813–839, 2007.

A. Tatarakis and I. Minis. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research*, 197(2):557–571, 2009.

D. Teodorović and G. Pavković. A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demands. *Transportation Planning and Technology*, 16(4):261–273, 1992.

F. Tillman. The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3 (3):192–204, 1969.

W. H. Yang, K. Mathur, and R. Ballou. Stochastic vehicle routing with restocking. *Transportation Science*, 34(1):99–112, 2000.

Chapter 4

Electric vehicle routing

4.1 Battery charging policies and charging function approximations

4.1.1 Full charging with constant charging times

Paper: Montoya et al.(2016a)

4.1.2 Partial charging with non-linear charging times

Paper: Montoya et al. (2016b)

4.2 Charging station capacity

Working paper: Froger et al. (2017)

4.3 Uncertain charging station availability

Working paper: Kullman, Goodson, and Mendoza (2017)

4.4 Shared charging infrastructure

Working paper: Kok et al. (2017)

4.5 Perspectives

4.5.1 Considering battery degradation costs

On going Ph.D. Thesis: Laura Echeverri

4.5.2 Multigraph formulations: trading off energy consumption and travel time

Future Post-doctoral project: Cagri Kok

4.5.3 Uncertain energy consumption

Future Ph.D. internship: Samuel Pelletier

Chapter 5

Industrial applications

5.1 Vehicle routing in a public utility

Paper: Mendoza, Medaglia, and Velasco (2009)

5.2 Technician routing with electric and conventional vehicles

Working paper: Montoya et al. (2017)

5.3 Maintenance scheduling in the wind industry

Paper: Froger et al. (2016a)
Working papers: Froger et al. (2016b, 2016c)

5.4 The pre-pack optimization problem

Book chapter: Hoskins et al. (2014)

Chapter 6

VRP-REP: the vehicle routing problem repository

Working paper: Mendoza et al. (2017)

Chapter 7

Conclusions and perspectives

Appendix A

Curriculum Vitae

Appendix B

Selected journal articles