

## The stochastic vehicle routing problem, a literature review, Part II: solution methods

Jorge Oyola<sup>1</sup> · Halvard Arntzen<sup>2</sup> · David L. Woodruff<sup>3</sup>

Received: 24 June 2016/Accepted: 21 October 2016/Published online: 2 November 2016  
© Springer-Verlag Berlin Heidelberg and EURO - The Association of European Operational Research Societies 2016

**Abstract** Building on the work of Gendreau et al. (Oper Res 44(3):469–477, 1996), and complementing the first part of this survey, we review the solution methods used for the past 20 years in the scientific literature on stochastic vehicle routing problems (SVRP). We describe the methods and indicate how they are used when dealing with stochastic vehicle routing problems.

**Keywords** Vehicle routing (VRP) · Stochastic programming · SVRP

### 1 Introduction

The vehicle routing problem (VRP) was proposed as a generalization of the traveling salesman problem by Dantzig and Ramser (1959), though the name used back then was “the truck dispatching problem.” Also the name “clover leaf problem” was suggested, because of the shape of a solution: a set of loops with a point in common. The term “vehicle routing” did not appear in the literature until the early 1970s (Eksioglu et al. 2009).

Several versions of the vehicle routing problem have been proposed, such as the capacitated vehicle routing problem (CVRP) (Toth and Vigo 2002). Different authors—e.g., Cordeau et al. (2002) and Toth and Vigo (1998)—have defined this version of the problem given an undirected graph  $G = \{V, E\}$ , where  $V = \{v_0, v_1, \dots, v_n\}$  is the vertex set, and  $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$  is the edge set.

---

✉ David L. Woodruff  
dlwoodruff@ucdavis.edu

<sup>1</sup> Institución Universitaria Esumer, Medellín, Colombia

<sup>2</sup> Molde University College, Molde, Norway

<sup>3</sup> UC Davis, Davis, CA, USA

$v_0$  represents the depot, and the other vertices represent the customers, each having a non-negative demand,  $q_i$ . The set  $E$  has an associated cost matrix  $c_{ij}$ , representing the cost of traveling from vertex  $i$  to vertex  $j$  ( $c_{ij} = c_{ji}$ ), in the symmetric case. A fleet of  $m$  vehicles with equal capacity  $Q$  is based at the depot. The solution to the problem is the one that minimizes the total routing cost, and the total demand of the customers in one route can not be greater than  $Q$ . Every customer is visited once by just one vehicle. Each tour includes the depot.

Several other versions can be found, depending on considerations and/or constraints included in the problem, e.g., the vehicle routing problem with time windows (VRPTW), vehicle routing problem with pickups and deliveries (VRPPD), vehicle routing problem with backhauling (VRPB), vehicle routing problem with distance constraints (DCVRP) (Toth and Vigo 2002).

To better describe the real world, stochastic data may be included in the model. In this case a new problem arises, the *stochastic (capacitated) vehicle routing problem* (SVRP or SCVRP). A tutorial with a synthesis of some recent literature can be found in Gendreau et al. (2014). A detailed description of the different problems found in the recent literature was presented in the first part of this survey (Oyola et al. 2016). Due to the complexity of the problem, considerable efforts have been put into solving the different versions of VRPs. Here, we attempt to provide a description of the different solution methods used in the latest formulations of the SVRP.

This survey proceeds in Sect. 2 with a summary of the various solution methods applied for solving SVRP variants. Tables summarizing the literature are presented in Sect. 3. Finally, Sect. 4 provides a brief summary of this work review.

## 2 Solution methods

Several approaches have been used for solving (or dealing with) different variants of the stochastic VRP. Here a review of these approaches is presented, trying to cover most of the methods designed to tackle such problems. The major dichotomy in terms of solution methods is of course between exact methods on one side versus heuristic methods on the other.

### 2.1 Exact methods

Although realistic stochastic VRP problems are often hard to solve, exact methods have been employed with some success. These methods view the problem as special case of an integer-, or mixed-integer program and employ some form of branching, so that eventually a probably optimal solution will be found. Even when employed on instances that are too large for full convergence, the methods can often find very good solutions.

### 2.1.1 Integer L-shaped method

The integer L-shaped method is a branch-and-cut algorithm that can be described in seven steps (Hjorring and Holt 1999; Laporte et al. 2002; Jabali et al. 2014). It can start using a previously found feasible solution (by another heuristic) as in Hjorring and Holt (1999) or with no initial feasible solution as in Laporte et al. (2002) and Jabali et al. (2014). General steps given here are those described in Laporte et al. (2002). The methods operate at each node of the search tree on a subproblem called the “current problem” (CP). Initially, it is the result of relaxing integrality, subtour elimination and capacity constraints in the original problem. In addition a lower bound  $\theta$  replaces the expected recourse cost in the objective function. CP is modified by adding constraints once violations are found.

1. The iteration count is set equal to zero. A new constraint,  $\theta \geq L$ , is added to the original problem, with  $L$  a lower bound on the expected recourse cost. The objective value of the best found solution is set to  $\infty$ . The only pendent node in the search tree corresponds to the initial current problem.
2. A pendent node is selected from the tree. If no pendent node exists, stop.
3. The iteration count is increased by one. The CP is solved, finding an optimal solution to it.
4. If there are any capacity or subtour elimination constraint violations, then at least one constraint is added. A lower bounding functional (LBF) may also be generated and added and the algorithm returns to Step 3. Otherwise, if the objective function of CP is greater than or equal to the objective function of the best solution found so far, the current node is fathomed and the algorithm returns to Step 2. LBFs are constraints that strengthen the lower bound of the recourse cost, and are associated with partial routes, which is to say: routes where a subset of customers are ordered in a particular way.
5. If integrality constraints are violated, branching is done on a fractional variable. The corresponding subproblems are added to the pendent nodes in the search tree and algorithm goes to Step 2.
6. The expected cost of the recourse is computed for the optimal solution of the current problem, and added to the objective function, instead of  $\theta$ . The objective function value of the CP is compared to the objective function value of the best found solution, from that comparison optimal solution to the CP can become the best found solution.
7. If the lower bound,  $\theta$ , for the expected recourse cost of the optimal solution to the CP, is greater than or equal to its actual expected recourse cost, the current node is fathomed and the algorithm goes to Step 2. Otherwise, an optimality cut is imposed and the algorithm goes to Step 3. The optimality cut just forces the algorithm to move to another solution that differs with the current in at least two edges not incident to the depot. The reason for that is that better solutions may exist since the lower bound  $\theta$  is strictly less than the actual expected recourse cost.

In Hjorring and Holt (1999) the L-shaped method was used to solve the single vehicle CVRPSD. Just one route is evaluated at the time, since the model is dealing with just one vehicle. A general lower bound for the expected value of the recourse action is imposed as optimality cuts. They use the concept of partial routes, where the same cut will improve the lower bound for all the solutions that share specific sequences of customers. The lower bound for a partial route is equal to the contributions from sequenced customers and bounds for exact stockouts (for the case of the discrete demand distributions) and normal stockouts in the subset of unsequenced customers. In general, the customers in the sequence are those near the depot, at the beginning and the end of the route. A subset of unsequenced customers is between them. A tighter lower bound was determined by considering just the customers at the end of the route. Starting from empty sequences, a greedy strategy is used to determine the partial routes. Two customers not included yet in the partial route are evaluated for insertion into it: the customer that according to the sequence in the current solution is closer to the depot at the beginning of the route and the one closer to the depot and the end of the route. The one that increases the value of the lower bound for the partial route is selected. The selected customer will have the same position, with respect to the depot, in the partial route and in the current solution. From all partial routes contained in the current solution, one is selected for use in the optimality cut to be added. The selection takes into account the lower bound value and the number of edges, where the idea is to have a small number of edges, but greater lower bound than the actual solution. Two versions of the algorithm are compared, one using specific optimality cuts and the other using cuts obtained from partial routes. The latter version managed to solve more problems to optimality and was generally faster. Instances with up to 90 customers and 105% of mean total demand computed as percentage of vehicle capacity, were solved to optimality.

The integer L-shaped method was used in Laporte et al. (2002) to solve the CVRPSD. The lower bounds are computed assuming that at most one stockout will occur per route. The problem of finding  $L$  is presented in a general way, for any probability distribution. It is then solved for demands that follow Poisson distributions and normal distributions. The LBFs are defined using the concept of *lower bounds of the expected recourse cost of partial routes*. In the routes where it exists, it is computed with a method similar to that used by Hjorring and Holt (1999), in the routes where that lower bound does not exist, it is computed as  $L$ , but using just the customers in such routes and the number of routes as the number of vehicles. The result is a constraint that establishes another lower bound for  $\theta$ . A separation procedure uses a heuristic to define the subsets of customers that will be treated as consecutive (two; one on each end of the route) and unstructured subsets, for each partial route. Instances with a number of customer varying from 25 to 100 vertices and a number of vehicles between two and four were solved to optimality.

An extension of the integer L-shaped method in Hjorring and Holt (1999) and Laporte et al. (2002) was used in Jabali et al. (2014) for solving the CVRPSD exactly. In this version LBFs are used to eliminate infeasible solutions. Applied to CVRPSD, the LBFs strengthen the lower bound of the recourse cost associated with partial routes found during the process. The construction of the LBF exploits the

information provided by partial routes and are developed based on the structure of the partial route. An exact separation procedure identifies partial routes and generates the corresponding bounds. A basic LBF implementation is compared with an integer L-shaped algorithm with no LBF. The version with LBF was able to solve more instances to optimality, and the run-times are less than half the ones used by the implementation with no LBF. The exact separation procedure generates better results than the heuristic used in Laporte et al. (2002). The LBF reduces the number of cuts added to the relaxed problem. The algorithm is able to solve instances with up to 60 vertices and four vehicles, and 80 vertices and two vehicles. The recourse cost is computed analytically.

The L-shaped method is also used in Chang (2005) to find solutions to a VRP with time windows and stochastic demands (VRPTWSD) was studied in Chang (2005). Recourse cost is computed analytically. A lower bound searching module is introduced, as well as a current solution calculation module. The method is tested on modified versions of Solomon's instances, with experiments performed on instances with up to 40 customers and four vehicles.

### 2.1.2 Integer L-shaped method with local branching descent

Local branching for the 0–1 integer L-shaped method is introduced in Rei et al. (2007). General principles of the algorithm are presented for solving integer stochastic problems, which are applied to the single vehicle CVRPSD. The main departure from the regular integer L-shaped method is a branching process when the optimal solution to a subproblem is not integer. The result is a method that tackles stochastic optimization problems with binary first stage variables. From a CP, two subproblems are obtained by adding constraints that divide the feasible space. One subproblem will have as a feasible space the solutions with no more than a given number of elements that are different from the binary elements in the optimal solution to the CP. The other subproblem corresponds to a larger feasible space; the solutions with more than a given number of elements that are different from the binary elements in the optimal solution to the CP. This larger feasible space can later be divided by adding more constraints of the same kind. When the smaller problem is not feasible, the parameter that indicates the maximum number of elements different from the optimal solution may be increased. Lower bounds are generated for each explored subregion of the feasible space. The standard L-shaped algorithm with partial route cuts (Hjorring and Holt 1999) is compared with two implementations of local branching, using cuts either locally or globally generated. Local branching with cuts generated locally outperforms the other implementations, solving instances with up to 90 nodes.

### 2.1.3 Branch-and-cut

A method that solves the deterministic equivalent of the stochastic problem was proposed to solve the VRP with stochastic travel times (Kenyon and Morton 2003). This method is applicable when the cardinality of the sample space is not large and it is proposed to deal with the two versions of the problem: the minimization of the

completion time and the maximization of the probability of being completed within a target time. However, the tests were done just for the first case. A probability of occurrence is assigned to each of the realizations of the stochastic parameters. The problem is then modeled as the minimization of the expected value of the objective function. For each possible realization of the parameters, a deterministic problem is solved. The method is based on branch-and-cut and it is applied to a model where the subtour elimination constraints are relaxed. The optimization problem is solved and if no subtours exist, the solution is optimal. If there are subtours not including the depot, a new solution is built joining each subtour with the main tour assigned to the same vehicle. The new solution is evaluated and if it is within a preselected percentage from the optimal solution, the algorithm stops. The percentage is computed using the objective function of the solution to the relaxed problem (lower bound) as a reference. If the new solution is not within the given percentage of the optimal solution, subtour elimination constraints are added and the procedure is repeated. The method was tested on four instances, each of which was a nine node network and had a fleet of two vehicles. Travel time follows a discrete distribution. The results were compared with the optimal solutions to the problem where mean values for the parameters are used. Solutions to the stochastic models (completion time) were better.

For the VRP with stochastic travel times, where the sample space is large, a method that integrates a branch-and-cut scheme in a Monte Carlo sampling procedure has been proposed (Kenyon and Morton 2003). In general, this method does not find an optimal solution. However, it is possible to bound the gap between the objective function of the solution found and the optimal value, with a confidence level. To do that, a lower bound and an upper bound are computed. An upper bound is computed (for the minimization of the completion time), taking a solution and computing the mean completion time, using a given number of scenarios. A lower bound is computed as the average of lower bounds estimators. Each of these lower bounds estimators are computed using a predefined number of scenarios. The total number of estimators (batch size) is also a parameter of the algorithm. Each lower bound estimator is found using a modification of the method that solves the deterministic equivalent of the stochastic problem, where subtour elimination constraints are added immediately after subtours are found. On each batch another upper bound is also used, the solutions already found in previous batches, are evaluated under the realizations of the actual batch and the smallest is selected. The method was tested on a 28 node non-completed graph, with two vehicles, and travel times were assumed to follow a uniform distribution. Service times were deterministic.

A similar approach as in Kenyon and Morton (2003) was used by Adulyasak and Jaillet (2016) to solve the robust and the stochastic approach of the CVRP with deadlines under travel time uncertainty. One of the main differences is how the cuts are added, while in Kenyon and Morton (2003) constraints are added upfront, in Adulyasak and Jaillet (2016) cuts are added iteratively when a feasible vector is found. Test results were compared to an iterative algorithm developed for the robust routing problem, similar to the classical Benders algorithm and to results obtained by Kenyon and Morton (2003). Performance of the algorithms is measured on

different stochastic problems. Travel times are assumed to follow different probability distributions, Triangular, Normal and Uniform. Also different number of deadlines are considered, at the last node, two nodes and all nodes. Depending on the problem, optimal solutions to instances with up to 80 nodes were found. The robust and the stochastic approach in general outperform the results by Kenyon and Morton (2003) and the iterative algorithm for the robust routing problem.

A branch-and-cut based VRP solver was used to solve the RVRP in Sungur et al. (2008). The robust formulation of the CVRP is obtained by replacing the constraint imposing capacity and connectivity in the original CVRP formulation. As a result the RVRP is more capacity constrained and may be infeasible even if the original CVRP is feasible, particularly in tight instances, where total demand is very close to the total capacity. The RVRP is solved as a deterministic problem and compared to the best solution to the original CVRP, using the cost of the total cost and the ratio of unmet demands as performance indicators. The comparison is done using randomly generated scenarios. On tests performed using standard problems, it was found that performance of the robust approach depends on the structure of the network, but still robust outperformed deterministic in several cases, particularly when the instances have clustered customers. Tests were also carried out on randomly generated clustered instances, finding that the robust approach performs better in instances with dense random zone around the depot. The robust solution was also compared to a strategy that uniformly distributes the excess vehicle capacity among all the vehicles. Solutions found using such strategy can have a lower ratio of unmet demand, but the cost might be higher than the robust solution.

A similar approach was used to solve a different version of the RVRP (Gounaris et al. 2013). The problem is solved using CPLEX 12.1 using cuts—called robust rounded capacity inequalities (robust RCI)—, which are satisfied by all feasible solutions to the RVRP, are added to the model. Violated cuts are identified using a variant of tabu search. In some formulations RCI are already contained. In these cases, the RCI are removed and dynamically reintroduced. It was found that the same set of routes is robust in each formulation. However, some of the formulations are more efficient than others, since less computational time is needed to solve them, notably two-index vehicle flow formulation and the vehicle assignment formulation (with RCI as subtour elimination). Most instances with less than 50 customers were solved to optimality. The optimality gap of the instances that were not solved to optimality (with less than 50 customers) is below 5%. The average gap on the other instances is 6.5%.

A branch-and-cut algorithm was used in Beraldi et al. (2015) to find solutions to the mixed capacitated general routing problem with probabilistic constraints (MCGRPC). Tests were performed on randomly generated instances, based on instances from the literature. The B&C algorithm was able to find an optimal solution in three out of 23 instances within the limited computational time (4 h).

#### 2.1.4 Branch-and-price

A new Dantzig–Wolfe solution method for the CVRPSD was proposed in Christiansen and Lysgaard (2007), based on a partitioning formulation of the

problem. The customer sequence for each route is known even when an integer solution to the problem is not. So the expected failure cost can be calculated before an integer solution is known.

Starting from a master program ( $P_M$ ), where the 0–1 integrality constraints are relaxed, partitioning constraints are changed to covering constraints, allowing more than one visit to the customers. Non-elementary routes are allowed (so not all elements are different in the route). The coefficient  $\alpha_{ir}$  (1 if customer  $i$  is visited in route  $r$ , 0 otherwise) is replaced by  $a_{ir}$  (number of times customer  $i$  is visited in route  $r$ ).  $P_M$  is initialized with  $n$  single customer routes. Solving  $P_M$ , a vector of dual prices is obtained. The dual prices are used in the search for columns with negative reduced costs. If the columns are found, they are added to the LP and the problem is re-optimized. The process is iterated until no more columns with negative reduced costs exist. At this point, the current solution is optimal for  $P_M$ . If the LP solution is integer and constraints are satisfied with equality, then it is also optimal for original problem. If constraints are inequalities, then they are changed to equalities and the LP is resolved, continuing with the iterative process. If the LP solution is fractional, then branching is done to eventually obtain an integer solution. The branch and price algorithm is described as a variant of branch and bound, where column generation is performed at each node in the branch and bound tree. Instances with up to 60 customers and 16 vehicles are solved to optimality.

A similar decomposition is used to reformulate the robust CVRP with deadlines and travel time/demand uncertainty (Lee et al. 2012). The problem is initially formulated as a path based set covering problem, where the decision variable is either to include or not to include a route in the solution. Integrality constraints are then relaxed, and the problem is solved with a restricted set of feasible routes, since the total number of feasible routes can be very large. In this context a feasible route is the one that meets the deadlines and capacity constraints at each customer, while most of the uncertain parameters are at their maximum deviations. The column generation subproblem is solved to find a column with negative reduced cost and when no column is found the procedure is terminated. A labeling algorithm (dynamic programming) is used to find robustly feasible routes with negative reduced cost. The existence of an optimal integer solution without cycles is not guaranteed. If the optimal solution has cycles, a branching procedure is applied. Solutions obtained by the robust model are compared to solutions obtained by a deterministic equivalent CVRP. A set of scenarios are generated and the tests estimate in which percentage of scenarios the solution feasibility is retained (robustness). In three different sets of instances, the robust model improves robustness by an average of about 48, 82 and 64% respectively, compared to the deterministic approach.

The same formulation as in Christiansen and Lysgaard (2007) was used in Gauvin et al. (2014) for solving the CVRPSD using a branch-cut-and-price algorithm. The bounds are computed by column generation solving a restricted  $P_M$ , with only a subset of all feasible routes, initialized by single routes (one for each customer). New routes are dynamically generated by adding columns (routes) with negative reduced costs, until no more such routes exist. If the solution is integer, then it is also optimal for the original problem. If it is not, violations of valid

inequalities are identified. If this is not possible, then branching is used. The generation of routes with negative reduced costs is done by solving a shortest path problem with resource constraints (SPPRC) as in Lee et al. (2012), executing a dynamic programming bidirectional labeling algorithm. The concept of *ng*-routes is used in combination with two-cycle elimination. Each customer has an associated set of customers with specific cardinality. A given route is prohibited from extension to include customers that belong to some of the sets associated with customers already in the route. A new dominance rule is included and it is considered to be valid when demands follow a Poisson distribution. This new rule makes it possible to eliminate more labels (partial routes) that cannot lead to optimal solutions. Solving the *ng*-route problem must be done at various nodes in the branch-and-bound tree. A tabu search heuristic is used to generate negative reduced cost columns, using moves that respect the current imposed branching. The search is restarted after reaching a number of iterations. Two types of cuts are dynamically added to accelerate the derivation of integer solutions. Capacity cuts imposing a lower bound on the number of vehicles required to serve a subset of customers, which is identified heuristically. Subset-row inequalities are used to prohibit the coexistence of routes that cover at least two customers in any triplet of customers. Violations to these inequalities are identified exactly. If still no violations in these constraints are identified, branching is used. Two options are available, branching on a sequence of two customers and branching on the number of arcs adjacent to a subset of customers. Both decisions are evaluated and one of them is selected. Results from this algorithm are compared to results in Christiansen and Lysgaard (2007). This algorithm solves 20 additional instances and manages to solve instances up to 86 times faster. Seventeen new instances are solved with up to 101 customers and 15 vehicles.

A similar approach was used in Taş et al. (2014b) for solving the CVRP with soft time windows and stochastic travel times. The problem is modeled as a set partitioning problem. A restricted  $P_M$  with relaxed integer constraints is solved using the routes of an initial feasible solution found as in Taş et al. (2013). The pricing subproblem of finding columns (routes) with negative reduced cost is modeled as the shortest path problem with resource constraints. If a new route with negative reduced cost is found, it is then added to the  $P_M$  and re-optimized. The pricing subproblem is solved using the algorithm proposed in Feillet et al. (2004) that extends a label correcting reaching algorithm (Desrochers 1988) including node resources, so the subproblem can be solved optimally. Initially, multiple visits to the customers are allowed, except for those in a given set  $S$ . If the optimal solution for the relaxed subproblem is integer, then it is also optimal for the original. Otherwise, the nodes that appear in the solution more than once are added to the set  $S$ . A new dominance relation is used for selecting the routes. Non-dominated routes with non-negative reduced costs and the dominated ones are added to a set called the Intermediate Column Pool (ICP). After re-optimizing the relaxed  $P_M$ , the reduced costs of columns (route) in the ICP are recomputed and columns with negative reduced cost are added to the relaxed  $P_M$ . The pricing subproblem is solved if no columns are found in the ICP. Routes stay in the ICP for a given number of iterations and its size is kept below a threshold. Branching is used if the optimal

solution to the relaxed  $P_M$  is not integer. During the tests, two stopping criteria were used, time and gap between best lower bound and best upper bound less than 5 %. Test were performed using two types of branching, Depth-First (DF) and Breadth-First (BF). It was found that BF provides better results. The algorithm is able to solve instances with up to 100 customers.

In Christiansen et al. (2009) a branch-and-price algorithm was proposed to solve the CARPSD where pricing is done by dynamic programming and an estimator of the total expected cost (lower bound) is computed analytically. The problem is modeled as a set partitioning problem, where the decision variables are the selection of routes to become part of the optimal solution. The problem is solved initially for a set of columns corresponding to the set of feasible routes, but the constraints related to the number of times that an edge must be serviced and the integrality constraints are relaxed. At each iteration the set of columns corresponding to the dual vector of constraints regarding the number of times the edge are visited, with an expected negative reduced cost, are added to the initial set of columns. The problem is iteratively re-solved (pricing) until no more columns with negative expected reduced costs are found. If the solution is integer, then it is optimal for the modified problem. If the solution is fractional, then branching and pricing is continued. An optimal integer solution to the modified problem is also optimal to the original one within some precision level, which depends on the quality of the lower bounds for the expected cost of the routes. The pricing problem is formulated as a shortest path problem and it is solved by dynamic programming. The branching rule is applied when an optimal solution to the modified problem has fractional values. The largest instance that was solved included 40 vertices and 69 service edges.

The VRP with hard time windows and stochastic service times was formulated as a set partitioning problem and solved by branch-cut-and-price algorithms in Errico et al. (2016). Valid inequalities are dynamically included at the end of the column generation procedure as a mechanism for tightening the lower bounds. Tests were performed on 56 instances, based on Solomon's instances. The algorithm is able to solve a few of the large instances (with 100 customers). Reduced size instances are obtained by considering the first 25 and 50 customers in the original instances (29 instances of each type). Tests show that instances with a larger support of the probability distribution are more difficult to solve. Regarding the two recourse actions, in the instances with smaller support of the probability distributions, obtained solutions are almost equivalent. However, the problem is harder to solve when it is formulated using *skip next* recourse. In instances with larger support of the probability distributions, the model including *skip next* recourse action has advantages, leading to improvements in the solutions' quality. The probabilities of the routes being feasible are computed analytically and the expected costs of the non-servicing penalties, this is done using labeling algorithms.

### 2.1.5 Other algorithms

An algorithm including a problem decomposition/collapsing method integrated with CPLEX was used to find solutions to a cash transportation vehicle routing and

scheduling problem under stochastic travel times (Yan et al. 2014). The main problem is decomposed into subproblems by separating it into several time periods. These subproblems are solved using CPLEX. At each iteration a new subproblem is added and optimized, keeping unchanged the departures and arcs selected in the previous subproblem. The solutions are evaluated analytically. A modified deterministic version of the model is presented. Both are compared by evaluating the best found solutions using scenarios. Tests are performed using data from a security carrier that serves the Chungli zone in Taiwan. The average gap between objective values and the lower bound is 1.32% for the stochastic version and 0.97% for the deterministic version of the problem. When the best found solutions are evaluated across the scenarios, as expected, the solution to the stochastic version of the problem has a better performance.

## 2.2 Heuristic approaches

There are many stochastic problems with no exact solution method known to work for reasonable sized problems. This creates the need for heuristics, which, even though they do not guarantee to find an optimal solution to the problems, may be able to find a good solution to them. This is a popular and growing area in the literature.

### 2.2.1 Adaptive large neighborhood search (ALNS)

An adaptive large neighborhood search heuristic was used to solve the capacitated arc routing problem with stochastic demands (Laporte et al. 2010) starting from an initial solution constructed by an algorithm called *stochastic path scanning*, which is a modified version of the *path scanning* (PS) method. At each iteration, one heuristic is selected randomly to destroy the current solution (removing  $q$  serviced edges), and then one insertion heuristic is selected randomly to repair the damaged solution, re-inserting the removed edges. The selection of  $q$  is also random. The acceptance of a new best solution is given by the *record-to-record travel* (RRT) algorithm (Dueck 1993). The total expected cost is calculated analytically. The efficiency of the algorithm is evaluated comparing the performance of the best found solution with the optimal solution to the deterministic version of the problem. Both solutions are evaluated and compared in stochastic simulation and the expected recourse cost is considered in both cases. Solutions found by ALNS show a better performance.

An ALNS heuristic was also used to solve the CVRP with stochastic demand and time windows in Lei et al. (2011). A modified version of the *push forward insertion* (PFI) heuristic is used to generate the initial solution. At each iteration,  $q$  vertices are removed by a removal heuristic, which is randomly selected. The solution is later repaired by an insertion heuristic. As in the previous case, the selection of  $q$  is random and the acceptance of a new best solution is given by the RRT algorithm. The objective function is to minimize the total expected cost, which is calculated analytically. The efficiency of the algorithm is compared against the best known

solution to the deterministic counterpart plus the expected recourse cost associated with it.

An adaptive large neighborhood search (ALNS) was proposed to deal with the CARP with stochastic service and travel times in Chen et al. (2014). A branch-and-cut algorithm for the CCP formulation of the problem was not able to find the optimum for some instances with 10 vertices and 25 arcs, within one hour. The initial solution is built by constructing one route at a time. Each route is constructed by adding the arc that has not been serviced yet and is closest to the end vertex of the current route. The distance is defined by the shortest path problem. During the search, the removal and insertion heuristics are randomly selected under the control of a weight that determines the selection probability. The weights are updated periodically during the search and depend on how successful each heuristic has been. Solutions are evaluated using the objective function for the SPR model. The algorithm terminates if there is no improvement after a certain number of iterations or the total number of iterations reaches a predefined value. Four removal heuristics are used: deterministic worst removal, random removal, reduce-number-of-vehicle removal and probabilistic worst removal. The four insertion heuristics used in the search are deterministic greedy insertion, probabilistic insertion, probabilistic insertion with recourse and probabilistic sorting insertion. The ALNS is used in both models, but in the case of SPR, the chance constraint can be violated. Experiments are conducted on instances with 5–25 vertices. For small instances with up to 10 vertices and 20 required arcs, tests were performed using the branch-and-cut algorithm and the ALNS. The branch-and-cut algorithm found the optimum in 31 out of 40 instances. ALNS found it in 30. The gap between the two methods for instances where optimum was not found range between 1.45 and 3.15%. For the instance with 10 vertices and 20 required arcs, the computer ran out of memory when using branch-and-cut. The ALNS was tested using both models. It was found that CCP requires the use of more vehicles. The SPR with recourse including the penalty for rescheduling services for the next day and also the excess duration of the work, shows better results, and it is suggested for real life application.

### 2.2.2 Dynamic programming

In contrast to stochastic programming, dynamic programming methods are typically used to find a policy rather than a one-time solution. By “policy”, we mean a function that maps the state of the system (e.g., the capacity remaining in a vehicle and the mean demand for stops remaining on a route) to an action (e.g., return to the depot).

A dynamic programming algorithm was used to solve the stochastic CVRP with optimal restocking (Yang et al. 2000). The problem posed was to find the optimal restocking policy, i.e., the right moment for a vehicle to return to the depot and restock before a route failure occurs. At every customer there are two options, to go back to the depot and restock or go on to the next customer in the route. Even though stockouts might occur, one of these two options is optimal. A solution is built using two different approaches: (1) starting with a single route through all the customers, obtained using a combined method (insertion + Or-opt), it is later

partitioned into small subroutines using dynamic programming. (2) By clustering first and then routing. Both approaches are compared with a lower bound obtained by solving the LP relaxation of a set partitioning formulation of the original problem. This was possible for instances with up to 15 vertices, for larger instances the two algorithms were compared to each other. The approach number 1 (route-first-cluster-next) shows better results when compared to the lower bounds; it found solutions within 2% of optimal. The same approach was found to perform better for larger problems. The results obtained were also compared against the solutions to the deterministic version of the problem, found by the Clarke–Wright savings algorithm (Clarke and Wright 1964). Average improvements up to 25% were observed.

Dynamic programming was also part of an algorithm used to solve the single vehicle CVRPSD with reoptimization (Secomandi and Margot 2009), formulating the problem as a Markov decision process (MDP). To overcome the computational challenges involved in solving large problems, just a subset of the states in the full MDP is considered, this methodology is called *partial reoptimization*. Starting from an initial sequence of customers, the tour is divided into sets/blocks where each block is re-optimized as a MDP. Two ways of dividing the blocks led to two different proposed heuristics. Given a parameter  $M$  and an initial sequence of customers, the blocks are formed with no more than  $M$  customers each. In the second approach, every block is formed with no more than  $2M + 1$  customers, in this case the same customer can be in several blocks. The performance of the heuristics is compared against the optimal reoptimization policy for instances with 10–15 customers. For bigger problems, it is compared with two rollout policies (Secomandi 2001, 2003) and an estimator for the lower bound, which is found solving the deterministic problem to optimality with unsplit delivery, i.e., if the demand of a particular customer is greater than the residual capacity of the vehicle, a trip to the depot for restocking is required before serving the customer. The proposed algorithms perform better than rollout policies and compared with the lower bound, the solutions obtained are on average within 10 and 13% for different type of instances.

A two-step rollout algorithm is developed for the VRPSD in Novoa and Storer (2009). In addition to the minimum cost at state  $x_k$ , a minimum is found for all the possible states  $x_{k+1}$  that can be generated from  $x_k$ . A cyclic heuristic is used. A dynamic programming recursion is used to compute the cost-to-go functions exactly. In addition, a Monte Carlo simulation is used to approximate the evaluation. To reduce the computation requirement an evaluation strategy is used. At a given state, if the probability of failure is lower than a given low threshold, just cases where the next customer is visited directly (without visiting the depot first) are considered. If the probability of failure is greater than a certain high threshold, the cases selected for evaluation are these where the next customer is visited after visiting the depot. Tests were performed on 120 instances, with a number of customers varying from 5 to 60. Compared with the method in Secomandi (2001) the two-step look-ahead method achieves improvements close to 5%. CPU time is reduced by the use of the evaluation strategy and the Monte Carlo simulation. Routing costs of solutions found by the Monte Carlo simulation are as good as

solutions found by exact computation. This was tested on 20 instances with 100 and 150 customers.

In Secomandi (2000), after formulating the single CVRPSD as a stochastic shortest path problem, the size of the resulting dynamic programming problem is found to be computationally prohibitive. To deal with that, a neuro-dynamic methodology is employed. Instead of computing the optimal cost-to-go for every stage, an approximation is used. In principle two similar policies are used. With *approximate policy iteration* (API), the cost-to-go values and pairs of states (training set) in a particular policy are found through simulation. Then least-square fitting is used to approximate the cost-to-go of the policy for all states, which is a vector of parameters. A new policy is found in a greedy fashion, together with a training set and sample cost-to-go values. The process is repeated, either until it converges or until a predefined number of iterations is reached. Convergence is not guaranteed. The other policy is the *optimistic approximate policy iteration* (OAPI), which is a variation of the previous policy. The training set is smaller and least-square problems are solved more frequently. The vector of parameters that define the cost-to-go of the policy for all states is found as an interpolation between the one used in the previous iteration and the one that would be used in the API policy. Only the results for OAPI are reported, since it always performs better. It is compared with a rollout policy, which performs over 7% better, attributed by the author to the fact that it uses an exact evaluation, and the OAPI uses an approximation in which approximation errors may lead to poor solutions.

A bilevel Markov decision process (MDP) is used to model the coordination between vehicles in the paired cooperative reoptimization (PCR) for the two-vehicle CVRPSD (Zhu et al. 2014). In the PCR strategy multiple customers can be assigned to one vehicle. The vehicles operate independently until information is shared, this is after any one of the vehicles finishes serving its assigned customers. The process starts dividing the customers into two groups, and assigns a group to each vehicle. When one vehicle has completed its assignment, the remaining customers (not served yet by the second vehicle) are divided into two groups and reassigned to each vehicle. This process is repeated until all customers are served. The sequence of the visits in each group is determined by partial reoptimization (Secomandi and Margot 2009). At each stage of the higher level (partitioning and communication), the remaining customers are divided into two groups. At the lower level the problem is formulated as MDP as in Secomandi and Margot (2009). The bilevel MPD can be solved by dynamic-programming backward recursion, and at each stage the expected cost-to-go values can be calculated. At a higher level all possible partitions are evaluated, using an approximation approach based on an a priori route that traverse all unassigned customers in a particular stage. The cost-to-go value is approximated using the DTD recourse action in the a priori route. Such an a priori route is generated by the rollout algorithm in Secomandi (2003). Two versions of PLC (Ak and Erera 2007) were implemented to be compared with the PCR: one with two vehicles and one more that finds the number of vehicle to achieve a given service level. PCR outperforms the other two approaches, with a cost improvement ranging from 20 to 30%.

### 2.2.3 Rollout algorithms

The single vehicle CVRPSD with reoptimization as recourse policy was solved using a rollout algorithm in Secomandi (2001). The customer that must be visited first is selected from a sequence of customers, previously found by the cyclic heuristic (Bertsimas et al. 1995). For each customer a new sequence is obtained. The customers will keep the order given in the initial sequence, however, in every new sequence a different customer will be selected as the first. The expected total length for each arrangement is computed and the first customer in the arrangement with the lowest cost is selected as the first customer in the route. Once the first customer is selected and visited (its demand becomes known) the next customer to be visited is selected. Two options are considered, the customer  $j$  that minimizes the expected value of the total length (computed as before) and the customer  $k$  that minimizes the expected total cost when visited after going to the depot for replenishment. The option with lower expected total cost is selected and the process is repeated until all customers have been served. The algorithm is described as based on neuro-dynamic programming/reinforcement learning methodology. The total expected cost of an *a priori* solution is computed analytically. Two experiments are carried out, one with small instances where the value of an optimal reoptimization policy is known, and a second experiment with large instances. Two versions of the rollout policy are analyzed in the first experiment. In the first version, in case of a route failure, the current customer is fully served by performing a trip to the depot and restocking before moving on to the next customer. In the second version at a given time, there may be more than one unserved customer whose demand is known but not yet served, so in case of a route failure, the customer may not be served immediately. Both versions generate near-optimal solutions. For the second experiment with large instances, just the first version of the rollout policy is tested, using three different *a priori* solutions; a nearest neighborhood heuristic together with 2-Int improvement steps, a cyclic heuristic enhanced by dynamic programming and the static rollout heuristic initialized by the tour produced by the former one. The best results are generated when using the solutions generated by the static rollout heuristic.

A rollout approach was proposed to deal with sequencing problems with stochastic routing applications in Secomandi (2003). As particular applications, the proposed algorithms are used with the TSP with stochastic travel times and the CVRPSD. A regular rollout algorithm is compared with iterated versions of it. Where the best found solution is later used as initial solution and the process is repeated. Tests were performed iterating the algorithm two and three times, obtaining better results in the latter version.

### 2.2.4 Local search

A tabu search heuristic was used to design delivery districts for the CVRPSD in Haugland et al. (2007). The districts are designed considering a long term perspective where they will stay fixed, to be used for various demand realizations. The routing every day is deterministic, since demand is known before vehicles leave

the depot. Districts are constructed so the expected travel cost within each district never exceeds an upper bound. A multi-start heuristic was also implemented and compared, but the tabu search heuristics performed better in the instances tested. The expected cost of the solutions is approximated using an upper bound.

An insertion-based solution heuristic called master and daily scheduler (MADS) (Sungur et al. 2010) was proposed to create the master plan to the VRPSTW with stochastic service times and probabilistic customers. A tabu search procedure is used to try to improve the solution. The solution process has two phases. Given a number of scenarios, in the first phase, a preliminary master plan and daily routes for each scenario are created. Two routing problems are considered, one for the master plan that serves high frequency customers using worst case service time and one for each scenario of daily schedule. The master plan is built using an insertion heuristic which works in a greedy fashion, inserting the cheapest of all feasible insertions. The cost of an insertion in the master plan is given by the increment in total time and TW penalty. The master plan is then used to construct daily schedules with a partial rescheduling recourse, for every scenario. Customers not present in the scenario are skipped and those not included in the master plan are inserted. Insertion in the daily routes incurs an additional cost, the reduction in similarity to the master plan. Daily schedules are improved with a tabu search heuristic. The second phase is iterative. Every scenario reports to the master plan which customers could not be inserted. Then the master plan prioritizes the unscheduled customers by the number of scenarios for which they could not be scheduled. Maximum priority feasible insertions are performed. Scenarios construct daily routes based on the updated master plan and the process is iterated until no further improvement is achieved. A buffer capacity is used in the first phase by reducing the latest time at the depot time window. It is used in the second phase to schedule additional customers. The solution obtained by the MADS is compared to a solution obtained by a algorithm that treats each scenario independently, which is called independent daily insertion (IDI). The objective function is to maximize the number of served customers and to minimize total time and time windows' penalties (weighted sum). Two real-world instances from UPS were solved by MADS and the solution is compared with IDI and the current practice. MADS is modified and results are also compared over consistent VRP (ConVRP) instances against the algorithm ConRTR proposed in Groér et al. (2009). Compared to IDI, MADS generates more similar routes at a cost of total time and total number of served customers, but the objective function shows a general better performance. When real-world data is used, solutions from MADS (given different levels of buffer capacity) are better than solutions from IDI and the current practice (obtained by a territory planning based routing algorithm). Compared with IDI, total time is increased, but similarity is improved. For the comparison over ConVRP instances, three criteria are used: total time, average arrival time difference and maximum arrival time difference. In average MADS performs better than ConRTR in all criteria.

The CVRP with stochastic travel times, soft time windows and service costs was solved using a tabu search heuristic in Taş et al. (2013). An initial solution is constructed by means of an insertion heuristic, then a tabu search heuristic is applied to the solution. A post-optimization procedure consisting of delaying the

dispatching time of the vehicles in each route is also applied. The solutions are evaluated through an analytic approximation of the expected cost. Different versions of the algorithm are tested, where different initial feasible solutions are used: solutions obtained by the insertion heuristic, with minimum total transportation cost; solutions obtained by the insertion heuristic, with minimum total weighted cost; and solutions obtained in the literature as optimal/best known for the deterministic problem. Better results are obtained by the algorithm when using initial solutions constructed by the insertion heuristic.

A location routing problem with disruption risk was solved by a two-step heuristic (Ahmadi-Javid and Seddighi 2013). During the first step, a solution is randomly built. In the second step there are two phases, location and routing. In each of those a simulated annealing heuristic is used. A 2-opt procedure is used to attempt to improve the routing. The quality of the solution is evaluated analytically. The heuristic was able to find optimal solutions found by CPLEX for small instances. It is also compared against lower bounds and to a different heuristic based on a solver for location routing problems. Optimal solutions are found, error bounds are relatively small and the location routing-based heuristic is outperformed.

Simulated annealing was also used to find solutions to the CVRPSD in Goodson et al. (2012). The main purpose is to show the potential of the cyclic-order neighborhoods using a heuristic with a simple structure. A cycle-order solution encoding is a permutation or ordering of the set of customers. Given a cyclic-order permutation  $\pi$  of the customers, a set  $\mathcal{R}(\pi)$  represents the feasible candidate routes consisting of contiguous elements of  $\pi$ . These candidate routes are generated using a sweep algorithm and their cost is computed analytically. The best solution that can be built using the routes in  $\mathcal{R}(\pi)$  is found by solving a set partitioning problem. Several cyclic-order neighborhoods are used. In the  $k$ -shift neighborhood structure the  $k$  contiguous elements, starting from selected index  $i$ , are moved to the positions immediately prior to a selected index  $j$ . In the reverse neighborhood, the order of the contiguous elements from index  $i$  to index  $j$  is reversed. The exchange neighborhood consists of exchanging the position of two elements. To reduce the computational effort, an updating procedure is used to obtain the set of feasible routes associated with one cyclic-order if the routes associated with a neighbor are known. The simulated annealing is executed in two phases. In the first phase, the classical deterministic CVRP is solved. The second phase attempts to improve the best found solution from phase I by taking into account the cost of recourse actions. Results were compared to Christiansen and Lysgaard (2007) where optimal solutions were reported for 19 out of 40 instances. The simulated annealing algorithm found 16 optimal solutions out of the 19 reported by Christiansen and Lysgaard (2007). In the 21 instances where Christiansen and Lysgaard (2007) did not find a provably optimal solution, the two phase procedure matches or improves the best found solution or the expected value of the best known deterministic solution.

In Goodson (2015), the cyclic-order simulated annealing procedure was modified to solve the multi-compartment vehicle routing problem with stochastic demands (MCVRPSD). A two-stage simulated annealing is used. In the first phase, solutions are evaluated by scenarios. While the second phase attempts to improve the solution by exactly calculating the quality of the solution. The method is compared with the

results reported by Mendoza et al. (2010) and Mendoza et al. (2011). Out of 180 instances, the cyclic-order simulated annealing procedure is able to improve the best known value in 159 instances and it matches the best known solution in 21 instances.

A tabu search algorithm was used to solve the CVRP with time windows and stochastic travel and service times in Li et al. (2010). No comparison is done with results obtained by other methods but the evaluation is done by estimating the expected values through Monte Carlo simulation.

A tabu search heuristic is proposed for the VRPSD with Pair Locally Coordinated (PLC) recourse (Ak and Erera 2007). The neighborhood structure is very similar to TABUSTOCH proposed by Gendreau et al. (1996), but moves are evaluated exactly, not approximately. This computation is done analytically, as the expected value of all the vehicles travel cost. The initial solution is found by a sweep algorithm. A service level parameter,  $p_\alpha$ , is set that is used to determine the number of customers in an a priori type II route. The probability that the vehicle serves all the customers in its route without a recourse detour before serving any of the customers of its paired vehicle must be greater than or equal to  $p_\alpha$ , which restricts the number of customers in the route. The number of customers in the type I route is also found using  $p_\alpha$ . The idea is that the probability of serving all customers in the paired tours, with the combined capacity and without any detours to depot, must be greater than or equal to  $p_\alpha$ . More vehicles are assigned to type I routes than to type II routes. The sweep algorithm assigns in a counterclockwise order, first the customers in a type I route, then those in the type II route. At the conclusion, each type I route will be operated in the counterclockwise direction and the type II routes are operated in the clockwise direction. The neighborhood of a solution  $x$ ,  $N(p, r, q, x)$ , contains a set of solutions generated by modifying routes as follows: (i) A set  $A$  of  $q$  customers is randomly chosen. (ii)  $\forall a \in A$  let  $B(a)$  be the set of  $p$  randomly chosen customers among the  $r$  nearest neighbors of  $a$ . (iii)  $\forall a \in A$ ,  $\forall b \in B(a)$ : remove  $a$  from its position and reinsert it immediately before or after  $b$  in  $b$ 's route. All the solutions generated by removing a customer from its tour and re-inserting it somewhere else. Here, each of  $q$  randomly selected customers is removed and reinserted, either immediately after or before one of  $p$  randomly selected customer neighbors from the set of its nearest  $r$  neighbors ( $q \leq r$ ). New partner tours can be created for unpaired vehicles and new tour pairs can be also generated. The PLC recourse strategy is compared with the DTD recourse strategy. Tests were conducted using instances with uniformly distributed customers and homogeneous demand distributions. In addition, a test was performed with real-world customer location data from grocery stores in Istanbul, Turkey, but using homogeneous demand distributions, as in the previous case. Results show better performance for the PLC recourse strategy.

Another example of the application of TABUSTOCH (Gendreau et al. 1996) is found in Erera et al. (2010), where it is used to find solutions for the vehicle routing problem with stochastic demands and duration constraints (VRPSD-DC). The solution method relies on solving an *adversarial* optimization problem, which determines a customer demand realization that maximizes the actual execution duration of an a priori tour, for checking if the maximum duration of a tour is

respected. The expected additional travel time due to recourse actions is computed analytically. The problem was tested against the unconstrained version CVRPSD in 54 instances. It was found that the number of vehicles required for serving the customers increased in 22 of the instances with duration constraints. An increment in the total expected travel time of more than 7% was observed in the small constrained instances. In general, however, this increment was relatively small. In some cases, as the fleet size increased, the total expected travel time decreased. This was more likely to be observed in large instances (60 or 100 customers) with large vehicle capacity.

A generalized variable neighborhood search (GVNS) was proposed to solve the VRP with stochastic service times in Lei et al. (2012). A Clarke–Wright algorithm is used to obtain an initial solution. The local search is driven by a variable neighborhood descent scheme where six neighborhoods are used, three inter-route and three intra-route. The algorithms proceed through the neighborhoods in a non-decreasing order by their cardinality. The search does not consider infeasible solutions. As part of the local search there is a granular search mechanism, which discards long edges and focuses on short edges that have a higher probability of appearing in high-quality solutions. The process is repeated following a variable neighborhood search scheme. A move is accepted applying the best improvement rule and as long as its value is within a certain threshold of the incumbent solution (best found solution in the current neighborhood). Solutions are evaluated analytically by means of a closed-form expression. GVNS was compared with variable neighborhood search and variable neighborhood descent, obtaining better results at a higher computational cost.

Five different metaheuristics were used in Bianchi et al. (2005) to deal with the single vehicle CVRPSD: simulated annealing (SA), Tabu search (TS), iterated local search (ILS), ant colony optimization (ACO) and evolutionary algorithm (EA). Tests are conducted to determine if using the a priori tour distance as an approximation of the objective function will generate better results than evaluating the solutions by means of dynamic programming recursion. It is found that EA, ILS and ACO perform better with the approximation. The TS and SA metaheuristics perform better with the evaluation by dynamic programming recursion. In general, the metaheuristics with better performance are EA, ILS and TS. An additional test is conducted where ILS and EA are hybridized with the 3-opt local search operator. While the local search is based on the a priori tour distance approximation, the acceptance and selection criteria are based on the dynamic programming recursion. Results are compared with the CYCLIC heuristic, and an iterated local search that uses a 3-opt exchange neighborhood, solving the problem as a TSP. Results show that hybridized versions of ILS and EA perform better.

A hybrid heuristic was proposed to solve the single vehicle CVRPSD in Rei et al. (2010), using exact algorithms (local branching) and Monte Carlo sampling. This Monte Carlo local branching algorithm follows a multi-descent scheme. Different solutions obtained from the TSP formulation of the problem are used as starting points, then local branching search phases are performed iteratively. The decision used for the branching criteria, and in that way limiting the feasible space in a subproblem, is fixing the maximum hamming distance relative of the solutions in

the feasible space to a given reference solution. At the first branching step, the optimal solution to the TSP, where the feasible region is limited to the unexplored region in previous descents is used as a reference. In the following steps it is replaced by the optimum solution to the previous subproblem and a fixed number of local branching steps is applied.

The recourse function is approximated using scenarios. The original recourse function is replaced by the approximation in the original problem, and the approximated problem is solved to optimality or until a specified time limit is reached. Each subproblem is solved applying the L-shaped branch-and-cut algorithm from Rei et al. (2007). Three types of cuts are generated by the algorithm, one of them is valid in all the subproblems, the others depend on the sample. A different sample is generated for each subproblem. At the end of the complete descent a number of solutions equal to the number of branching steps are produced. To identify the best one, each of them must be either evaluated using the routing cost and the actual recourse function, if possible, otherwise more sampling is required to produce an estimator of the expected total cost for each of the solutions. The procedure is repeated a certain number of times.

Comparisons were done against results from L-shaped branch-and-cut algorithm (Rei et al. 2007) and the Or-opt algorithm used to build a initial single route in a Route-First-Cluster-Next algorithm (Yang et al. 2000). Computational experiments show that the algorithm is able to obtain better results than the Or-opt algorithm and same results as L-shaped algorithm, but using less computer time.

The probabilistic multi-vehicle pickup and delivery problem (MPDP) was solved by a local search algorithm in Beraldi et al. (2010), where an insertion algorithm is used to build the initial solution. The efficient neighborhood search takes advantage of the previous computation of the expected objective function, which is done analytically. Moves are evaluated taking a vertex from a tour and moving it a position forward or backwards, up to a certain level. Another option is removing a vertex from a tour and inserting it at the first position of another tour. At every move not all the neighbors' solutions have to be evaluated, just the cost should be updated (analytically), depending the move that is performed. Equity of workload among vehicles is computed as percentage deviation of the total cost. This aspect is just calculated and results presented, but not considered as a constraint or extra objective in the problem. Two ways of evaluating the neighboring solutions are tested. In one, the expected cost is computed from scratch, in the second, the efficient neighborhood search is used, which is shown to be computationally more efficient.

For the construction of the planned and operational routes in the CVRPSD with time windows, different local search heuristics are used in Erera et al. (2009). The planned primary routes are constructed by an insertion heuristic, which starts with  $m$  routes serving  $m$  seed customers, and the rest of the customers are inserted one by one. Local search is used during the process to improve the partial solution. During the construction phase, the capacity constraints are checked, trying to keep the probability of the solution being feasible greater than or equal to a given parameter. This is done analytically. The assessment of the time window constraints is done by random sampling, the probability of satisfying the time windows constraints is estimated and a solution is considered to be time window feasible if the probability

is above certain value. During the insertion, two aspects are considered regarding the quality of a solution, the expected travel time and expected duration (may be different due to waiting times). A local search improvement procedure is applied. There is a vehicle reduction procedure that tries to eliminate the routes with the shortest average route duration. For the construction of the planned backup routes, demand scenarios are generated. Backup routes are not actual routes, since customers assigned to them are not sequenced. For each scenario, customers not placing orders are skipped. If all routes are feasible, no information is obtained and the next scenario is generated. If one or more routes are infeasible, then a customer is selected randomly to be removed from the route and reinserted in the route that minimizes the change in route quality. The process stops when the solution is feasible or when a given number of moves have failed to make it feasible. If the solution is infeasible, then no information is obtained, but if feasible, a local search procedure attempts to improve the solution and then irregular customers are inserted into the solutions, starting with the ones furthest from the depot. For every scenario the route serving regular customers is recorded. Vehicles with the highest count, not including the primary vehicle, are selected as backup vehicles.

The construction of the delivery (operational) routes follows the same strategy as selecting the secondary routes. But for achieving feasibility, customers can be moved from their primary route to their backup route. Service times are taken into consideration in this process. Several aspects determine the quality of a solution if a feasible operational set of routes can be produced: total travel time, duration of all routes, number of vehicles used, number of customers visited by primary vehicle. The results of using the methodology are compared with historical data. Two days of the week are compared: Thursdays, with the highest demand, and Mondays, with the lowest demand. For Thursdays, the proposed delivery routes are shorter and, in addition, three fourths of historical routes are infeasible. The proposed delivery schedules reduce the number of miles and the travel time. If the actual requirement of the customers being visited by just a primary or backup driver is dropped, and instead of that, any driver is allowed to visit any customer, the average total miles is reduced by 4%. For Mondays, the improvements are more evident. Fewer routes are used. There is a cost connected to serving a regular customer just with a primary or backup driver since total miles will be reduced by more than 8% if more drivers are allowed to serve such customers.

The problem of package delivery with driver learning was solved in two steps by Zhong et al. (2007). In the first step the strategic decision of designing the core areas is taken. In the second step, the cell routing is done. Here a tabu search heuristic is used to solve an assignment problem whose objective is to minimize the cost of assigning the cells to core areas. This tabu search is allowed to visit the infeasible region, and in addition to simple moves, compound moves are also allowed. The probability of serving a core area within the duration of a work shift is required to be below a certain threshold. The operational routing is done by first routing the cells within the core areas and later adding the rest of the cells to the partial routes, at the lowest cost. More routes are added if needed and the learning curve model is introduced. Cells closest to the depot are not assigned to any core zone. A fixed number of core areas is used, which is taken from historical data and is equal to the

minimum number of drivers used over a certain period. When designing the core zones, the objective is to minimize the expected total time (service and travel time) used to satisfy the demand of cells in the core areas. Cells not assigned to any particular core area are also taken into the objective as part of one additional core area with a lower learning curve and higher service time. Expected total time is approximated analytically. The operational routes are built using a deterministic routing algorithm that incorporates, instead of a single customer, the cell concept and the drivers' learning curve. The performance of the tabu search heuristic is compared to a lower bound found by solving to optimality a problem where the nonlinear constraints have been replaced by linear ones. On average, solutions found by the tabu search are 3% above the lower bound. The approach using core areas is tested against a policy of deciding a different routing every day. It is assumed that drivers will not reach maximum learning level on this new policy. On average the core area model uses 4% fewer drivers and its total duration time is 4% less. If in the non-core area model drivers are allowed to get the maximum learning level, the solution obtained represents a lower bound. On average the core area model uses 6% more vehicles, total duration time is 7% longer and total distance is 5% greater, compared to the lower bound, which is considered to be good performance.

A local search heuristic was used to solve the CVRP with soft time windows and stochastic travel times (SCVRPSTW) in Russell and Urban (2008). The solution procedure is a tabu search with three phases. First, an initial solution is obtained, which is improved in the second phase by a tabu search. In the last phase a postprocessing procedure is used to optimize the waiting times before visiting each customer. The initial solution is built using a deterministic tabu search with a mixed neighborhood search procedure that uses node exchanges and edge exchanges. Waiting time is allowed at the depot, but not at customer locations. Once the initial solution is obtained, a tabu search heuristic that evaluate moves analytically is applied to it. This heuristic uses basically two types of moves. One move is to remove a node from its position in route  $r_i$  and insert it in route  $r_j$  ( $r_i$  may or may not be different from  $r_j$ ). The second move is to swap the position of two nodes. As a diversification strategy, the tabu tenure is increased if solutions are found to be repeating too often. In case it happens extremely often, the search is diversified by making random moves. The post-optimization tries to find optimal waiting values before each customer.

The objective function used to guide the search is a weighted sum of the objective functions. Tests are performed using two different objective functions,  $1000V + 0.5D + 0.5P$  and  $1000V + D + 0.2P$ , with  $V$  equal to the total number of vehicles,  $D$  the total traveled distance and  $P$  the penalties associated with servicing customers outside their time windows. Expected penalties are computed in closed form. Results are compared with the solution obtained in the first phase. When the first objective was used, the stochastic travel time approach was able to reduce the number of vehicles in 10 out of 16 instances. The total distance was reduced in 11 instances and the time windows penalty was reduced in 12 instances. If more priority was given to the total travel distance than to the expected penalty, as in the

second objective, solutions with fewer vehicles and less distance traveled are obtained, however, the time windows penalty increases. For this case the stochastic travel time approach was able to reduce the number of vehicles in 12 out of 16 instances, the total distance was reduced in 14 instances and the time windows penalty increased in 12 cases.

Solutions to a VRP with stochastic time-dependent travel times were obtained by a tabu search algorithm in Lecluyse et al. (2009). A  $\lambda$ -interchange is used as a neighborhood structure, where up to two customers are exchanged between two routes ( $\lambda$  is set to be equal to two). The times are approximated and the simulations show that total travel time of a route follows a lognormal distribution, when the travel times between two customers is also lognormal. Variances are determined heuristically (approximated). A tradeoff solutions set between the expected travel time, the standard deviation and the 95th-percentile of the travel time, is presented.

An iterated tabu search (ITS) heuristic was developed in Zhang et al. (2013) to solve the stochastic vehicle routing problem with soft time windows under travel and service time uncertainties. Arrival time distributions are generated using an approximation method called  $\alpha$ -discrete. The ITS is basically a tabu search embedded into an iterated local search procedure. It includes a route reduction mechanism that attempts to reduce the number of required vehicles. A record of move costs is kept to avoid repetitive computation, so at every iteration just the possible moves from/to the two routes edited in the previous iteration are computed. Tests were performed on variants of Solomon's instances (Solomon 1987). The accuracy of the  $\alpha$ -discrete approximation method is compared against the method in Chang et al. (2009), which was used for the stochastic dynamic traveling salesman problem with hard time windows. A smaller estimation error was obtained by the  $\alpha$ -discrete method. The reference values for the comparison were obtained by stochastic simulation in Li et al. (2010). Tests were performed using different parameters settings, as a result a tradeoff solutions set between mean travel time and tardiness penalty cost is obtained.

A tabu search algorithm was used in Taş et al. (2014a) to find solutions to a time-dependent VRP with soft time windows and stochastic travel times. Two operators were used, one that changes the location of a customer within a route, the other removes a customer from a route and inserts it into another. A post-optimization procedure is applied to each route; it shifts the starting time (time leaving the depot) 10 min. This is done repeatedly until the total cost of the route is no longer improved. The algorithm was tested using Solomon's instances (Solomon 1987) and the results were compared against the performance of an implementation of an adaptive large neighborhood search (ALNS). When service times are assumed to be equal to zero, tabu search gives better results in clustered instances, random instances with tight time windows and randomly clustered instances with tight time windows. For the cases where the service times are assumed to be greater than zero, tabu search gives better results in clustered instances and random instances with tight time windows. In instances with 200 customers, tabu search performs better, but it takes a longer time. Simulations are run to assess the quality of the estimations of arrival time, lateness and earliness. The differences were at most 3.27%. The ALNS algorithm was embedded into a simulated annealing framework, where an

improving move is always accepted. A non-improving move is accepted with a probability that depends on the quality the move and a temperature parameter that decreases along the search. The same post-optimization procedure was applied, as before. For the cases where the service times are assumed to be greater than zero, ALNS gives better results in randomly clustered instances and random instances with large time windows. In instances with 200 customers, ALNS is faster, but at the cost of lower quality.

In Ehmke et al. (2015) a tabu search heuristic was implemented to find solutions to a VRP with time windows and stochastic travel times. The algorithm is the LANTIME tabu search that was introduced in Maden et al. (2010). At each iteration, the neighborhood operators are randomly selected from a list (adapted cross exchange, insertion/removal, one exchange and swap). Tests were done using Solomon's instances. Distributions of arrival times were approximated and simulation was used to evaluate the approximations. In all cases the mean is within one percent of the simulated value. The mean absolute percent error (MAPE) is within one quarter of the one percent, for normally distributed travel times. The percent error increases when the skewness of the travel time distribution increases (normal—gamma—exponential). Numerical experiments were performed using both normal and gamma distributions for the travel time.

Several local search heuristics are proposed to find solutions to the robust CVRP with uncertain travel times (RVRP) in Solano-Charris et al. (2015). A Clarke and Wright heuristic for the RVRP, starting with a trivial solution where each customer is visited by a different vehicle and routes are merged in a variety of ways. A randomized version is also used where before merging two routes, a perturbation is randomly computed. A local search procedure is employed that uses intra-route and inter-route moves: relocate (different versions), interchanges, 2-opt (two versions). Four other metaheuristics are also used: GRASP, iterated local search (ILS), Multi-Start ILS and Multi-Start ILS alternating between two search spaces, TSP tours (giant tours) and RVRP solutions. This search works on a pair  $(\omega, \tau)$ , where  $\omega$  is a RVRP solution and  $\tau$  is a giant tour obtained concatenating the routes in  $\omega$  (no copies of the depot are included). If  $\tau'$  is obtained by perturbing  $\tau$ , a *split* process is used and a solution  $\omega'$  is obtained from  $\tau'$ . After applying local search to it, it is compared to  $\omega$ . If found to be better, the latter is updated and a new giant tour is obtained. The heuristics were tested on two sets of randomly generated instances. The first set consists of 18 small instances, with 10–20 customers and 2–3 vehicles. Results from these instances were compared to solutions to the MILP formulation of the problem, obtained by GLPK. The second set consists of 24 instances with 50–100 customers and 5–10 vehicles. Ten out of eighteen instances were solved to optimality by GLPK. The heuristics are very close to GLPK but faster. All ten proven optima are found and three upper bounds are improved. For the larger instances, the tests show that the best metaheuristic is MS-ILS with giant tours, followed by MS-ILS, ILS and GRASP.

A neighborhood search algorithm was proposed for the mixed capacitated general routing problem with probabilistic constraints (MCGRPC) in Beraldi et al. (2015). It has three main elements: a diversification strategy, intensification phase and an exact local search. Diversification may allow the search to explore more

promising areas of the feasible space. In the intensification phase, several neighborhoods are used following a Variable Neighborhood Descent (VND) strategy. In this phase, moves leading to infeasible solutions are accepted as a strategy for escaping from local optima. The exact local search is a route optimization that uses the B&C algorithm to improve portions of the solutions. It is also used to recover feasibility, selecting a subset of routes, optimizing them and keeping the rest of the routes fixed. Tests were performed on randomly generated instances, based on instances from the literature. Results were compared to a B&C algorithm. The proposed method outperforms B&C, especially in instances with more than six customers.

In the framework of the vehicle routing problem with hard time windows and stochastic travel and service times, an iterated local search procedure is used in Miranda and Conceição (2016), mainly to find solutions where a method employed to approximate the service levels (probability of arriving before  $l_i$ ) is evaluated. Such a method is a statistic model that calculates the cumulative probability function of the arrival time as a sum of a truncated non-normal with a normal variable. It is compared to the  $\alpha$ -discrete method (Zhang et al. 2013), using the accuracy of the estimates for the service levels and the computational times as performance indicators. The error of each method is obtained from the difference between their results and results obtained by stochastic simulation. The proposed method produces slightly better results and is about 3.6 times faster.

### 2.2.5 Constructive algorithms

The multi-compartment CVRPSD was solved by three different constructive heuristics in Mendoza et al. (2011). One of them is a stochastic Clarke–Wright heuristic, which starts from a solution with round trips to every customer and merges the routes that will have a better impact on the total expected cost. The two other heuristics are different approaches of a look-ahead heuristic with two steps. In the first, the traveling salesman problem is solved and in the second a clustering procedure is performed. The preferred iterative look-ahead technique (pilot method) (Voß et al. 2005) is used to avoid suboptimal moves of the greedy heuristics used in the routing step (nearest neighbor and nearest insertion). The difference between the two different approaches is given by the heuristic used in the routing step. For the clustering step a dynamic programming algorithm is used. The total expected cost is calculated analytically. A stochastic 2-Opt procedure is applied as post-optimization to the three heuristics, it evaluates the moves using the deterministic values, and just promising moves are evaluated in terms of the stochastic values. Results from the algorithms are compared with the results obtained by a memetic algorithm (Mendoza et al. 2010), where tests show that even though the quality of the solution is not necessarily better, the algorithms are much faster. The algorithm is also tested for the CVRPSD and compared with results reported by Christiansen and Lysgaard (2007). The two approaches of the look-ahead heuristic algorithms were able to find some new best known solutions, and for instances with a reported optimal solution, the three algorithms were able to find a good solution in short time.

In Juan et al. (2011), the CVRPSD was solved by a multi-start search procedure, combined with the Clarke–Wright heuristic, which the authors report to have behavior similar to GRASP (M.G.C. Resende 2010). However, the methodology used to solve the problem can be applied with any efficient algorithm for the CVRP. The strategy is to use part of the vehicle capacity as a safety stock, while the remaining capacity is used during the routing step. The CVRPSD is solved as a CVRP and the vehicle capacity is set to the actual capacity minus the safety stock. Once the solution to the CVRP is obtained, simulation is used to evaluate it. The reliability of the solution is also computed as the probability of not having failures. Although no comparison is done with other methods, the algorithm is tested on different types of instances. The only comparison is with the best found solution to the deterministic CVRP counterpart of the problem, but measuring the performance of the solution in the stochastic framework.

A multispace sampling heuristic (MSSH) is proposed by Mendoza and Villegas (2013) to find solutions to the CVRPSD. It follows a two phase solution strategy. In the first phase, it samples multiple solutions. In the second phase it uses the sampled elements to build a solution. It combines a set of randomized heuristics for the TSP, a tour partitioning procedure and a set partitioning model. In phase one it uses a randomized TSP heuristics to build a sample of giant tours (TSP-like). From each tour, every feasible route that can be extracted without changing the order of the customers is added to a set of routes. The objective function value of the best found solution is used in phase two as an upper bound. In phase two a set partitioning formulation of the problem is solved and a solution is assembled using the routes built in phase one. The sampling heuristics are randomized nearest neighbor, randomized nearest insertion, randomized best insertion and randomized farthest insertion. Results are compared to Christiansen and Lysgaard (2007), Mendoza et al. (2011) and Goodson et al. (2012). Instance size ranges from 16 to 60 customers and the demand follows a Poisson distribution. Goodson et al. (2012) is able to find more best known solutions, but MSSH is more stable, since it finds solutions close to the best known solution more often.

A combination of GRASP and a heuristic concentration (HC) is proposed to find solutions to the CVRPSD in Mendoza et al. (2015). The GRASP uses a set of randomized route-first, cluster-second heuristics to generate starting solutions. A variable neighborhood descent procedure is used for the local search phase. A starting solution is constructed greedily using a randomized TSP heuristic that builds a giant tour; then a split procedure is used to get a feasible solution. A VND procedure is later applied to the solution. Once a local optimum is found, its routes are added to a set of routes to be used later by the HC. The route-first heuristics are similar to those used in Mendoza and Villegas (2013). The VND has two neighborhoods, relocate and 2-opt. The evaluation process of the moves has three steps. First it checks the affected routes for feasibility. If not feasible, the move is discarded and not evaluated. If feasible, the second step evaluates the deterministic part of the objective function. If the moves leads to a degraded solution (deterministic), then it is discarded. If the move improves the deterministic evaluation of the objective function, the third step consists of analytic evaluation of

the objective function. The heuristic (GRASP+HC) was tested on 40 instances of the classical CVRPSD from Christiansen and Lysgaard (2007) where demand follows a Poisson distribution. The heuristic is able to find all the 40 best known solutions, thus outperforming state-of-the-art metaheuristics (Goodson et al. 2012; Mendoza and Villegas 2013) that did not succeed in finding all of them. The instances for the CVRPSD with duration constraints were built by adding duration constraints to 39 of the instances in Christiansen and Lysgaard (2007). Solutions found using the GRASP+HC were compared to the solutions to the classical CVRPSD. Using 0.05 as the maximum probability of violating the duration constraint, just 3 out of 39 solutions to the classical CVRPSD remain feasible. On the other hand the solutions that fulfill such constraint found by GRASP+HC increase the cost by 2.10% in average. The penalty formulation was computed in three different ways, linear, piecewise linear and quadratic. The performance of the best known solutions to the CVRPSD, measured as total expected overtime and the total expected overtime cost, was compared to the performance of the solutions found by the penalty formulation of GRASP+HC. The linear penalty reduces the expected overtime by 52.83%, the piecewise linear by 75.51% and quadratic by 93.52%. There is an increment on the expected duration of the routes; on average it increases 0.79, 1.89 and 4.79% respectively.

As a way to illustrate the advantages of modeling stochastic time by phase-type (PH) distributions, Gómez et al. (2016) uses a multispace sampling heuristic (MSSH) (Mendoza and Villegas 2013) to deal with the distance-constrained CVRP with stochastic travel and service times (DCVRPSTT). This selection was done for convenience, since MSSH explores the same areas of the solution space independently of the route evaluator being used. A route evaluator using PH distributions and one based on normal distributions and Monte Carlo simulation are embedded into an adaptation of MSSH employing a two phase solution strategy. In the first part it samples multiple solutions. In the second phase it builds the best possible solution using parts from the sample solutions. Instances were adapted from the literature and for each instance three scenarios are assumed to have a different probability distribution for the travel time in each scenario: Erlang, lognormal and Burr distribution. The service level is the same in all cases. Two alternatives are considered for service time: deterministic and exponentially distributed. When travel time is modeled with an Erlang distribution, it was found that the normal route evaluation is the best choice, and the simulation route evaluation, the worst. When travel time is modeled with a lognormal distribution, the PH route evaluator finds better routes, but is more computationally expensive. When travel time is modeled with a lognormal distribution, the PH route evaluator finds better routes, and normal route evaluation is not able to find feasible solutions. In general it was found that the Normal route evaluator is a good option when the travel times do not have a large skewness. Monte Carlo simulation leads to overly optimistic solutions that do not satisfy the chance constraints. The PH route evaluator leads the algorithm to find solutions with similar quality as Monte Carlo simulation, but more reliable.

### 2.2.6 Progressive hedging

In Hvattum et al. (2006) a dynamic CVRP with time windows and stochastic customers and demands was solved using a sample scenario hedging heuristic, called dynamic stochastic hedging heuristic (DSHH), based on progressive hedging (Rockafellar and Wets 1991). Sample scenarios are used to guide the heuristics that build a plan for each time interval. Each scenario is solved as a deterministic CVRP using a simple insertion heuristic, common parts are used to build a solution to the dynamic and stochastic problem. The algorithm is compared against two different methods, a local search heuristic that solves the deterministic problem and a myopic dynamic heuristic that solves the problem at each stage ignoring the stochastic information. DSHH was able to find solutions with a travel distance more than 15% shorter than the myopic dynamic heuristic.

### 2.2.7 Evolutionary algorithms

A genetic algorithm is used in Ando and Taniguchi (2006) to solve the CVRP with soft time windows and stochastic travel times. However, a detailed description of the algorithm is not presented. The quality of the solutions is evaluated using simulation. The best found solution is compared against the usual operation, for 5 days. The solution found by the algorithm performs better. The average of the total cost was reduced by about 4%, and its standard deviation was reduced by about 75%, i.e., more reliable routes.

The multi-compartment CVRPSD was solved using a memetic algorithm in Mendoza et al. (2010), where the solutions obtained by a genetic algorithm are improved by two local search procedures: relocate and 2-opt. The reparation and fitness evaluation of the individuals in the population is performed in an analytic way through *stochastic split* (s-split), which is an extension of the Prins algorithm (Prins 2004). The results obtained by the memetic algorithm were compared with the stochastic Clarke–Wright, where all possible merges are recalculated at each iteration, and with a spare capacity strategy, which consists of solving the deterministic version of the problem (using a memetic algorithm in this particular case), but preserving some free capacity on each compartment. The proposed memetic algorithm is shown to find better solutions; however, it is more time consuming. The algorithm was also tested on a set of instances for the multi-compartment CVRP, and the results were compared against a tabu search and a memetic algorithm reported by Fallahi et al. (2008). Some new best solutions were obtained and on average, results show a gap of about 1% with respect to the best known solutions.

A multi-objective CVRPSD was solved using a multi-objective evolutionary algorithm in Tan et al. (2007). The quality of a solution is computed by means of a route simulation method, where several sets of demands for all customers are generated. The solution is evaluated on each set and an average value is obtained for each objective. The obtained averages are used to rank the solutions based on the Pareto dominance concept, using for this purpose the expected values of the objectives.

In Zhang et al. (2012), the stochastic travel time CVRP with simultaneous pickups and deliveries is solved using a scatter search heuristic, with the initial set of solutions created using a variant of the Clarke–Wright algorithm. In this problem the chance constraints (time limit constraints) are transformed into fixed constraints and the problem can be solved as a deterministic problem. The results are compared against results obtained by a genetic algorithm, the proposed algorithm produces solutions that are on average up to 13% better than the genetic algorithm.

In Sørensen and Sevaux (2009), a memetic algorithm with population management was used to deal with robustness and flexibility of the CVRP with stochastic demands and travel cost, and the VRP with stochastic customers, respectively. The algorithm consists of a genetic algorithm hybridized with two versions of tabu search that are used alternatively. The diversity of the population is controlled with a distance measure. This allows the population to be small, but keeps it diverse. The edit distance that measure the number of steps (add character, remove character, substitute character) that would be performed in one solution to become another, is used to measure the average distance between a solution and the population. The distance is required to be above a certain threshold, before the solution is added to the population. The threshold is decreased to intensify the search and it is increased again when the search is stuck in a local optimum. The two tabu search heuristics are insert tabu search, and swap tabu search. The former attempts to insert any customer at any other tour and the latter attempts to swap any pair of customers in the solution. Robustness evaluation was used in the binary tournament of the genetic algorithm (by simulation), however, it was not used in the tabu search procedures, where the deterministic objective function is used to select the next move. It was found that robust solutions to the CVRP with stochastic demands and travel cost will have a good deterministic objective function value, the reverse is not always true. For the CVRP with stochastic customers, it was found that the robust approach is not profit maximizing, since the best solution to the deterministic problem, when all customers require the service, is also likely to be the best solution to the problem with a reduced set of customers.

The SCARP was solved by a process consisting of two parts (Fleury et al. 2002, 2005b), optimization and robustness evaluation. The optimization is done by a genetic algorithm that uses a local search as a mutation operator, designed for solving the deterministic CARP (Lacomme et al. 2001), but not all solutions are subject to local search. The algorithm stops after a specific number of iterations, or after a certain number of iterations with no improvement or when reaching a lower bound. If the lower bound is not reached in the main phase, several short restarts are performed with a higher probability for the solutions to be subjected to a local search procedure that performs best-improvement moves and stops when no more improvements are available. Local search moves include removal of one or two consecutive edges (that require service) from a route, and reinsertion in another position, exchange of two edges, and 2-opt moves. The robustness evaluation is performed on the best solution found by the optimization process. It consists of a statistic evaluation of performance indicators such as the average cost, average number of trips, percentage of solutions requiring extra trips to the depot, standard deviation of the total cost and standard deviation of the total number of trips. For

statistical purposes, the solution is evaluated in several independent scenarios. The optimization part is done using three different approaches, *tight* and *slack*. In both cases, the deterministic problem is solved using the expected values of the demands as parameters. In the *slack* approach, just a percentage of the vehicle's capacity is utilized when making routing decisions. The *slack* approach shows reduction of the variability when compared with the *tight* approach.

In addition to the *tight* and *slack* approaches, a memetic algorithm where the objective function is computed analytically was used in Fleury et al. (2004), a similar approach was called *law* in Fleury et al. (2005a). In the *law* approach, two objective functions are considered to be minimized: the average cost of the solution and the average cost plus a fixed constant multiplied by the standard deviation of the cost. Even though different instances were used in Fleury et al. (2005b, 2005a), the results were similar. The *law* approach, when the average cost is minimized, produces solutions with less variability than the *tight* approach, but not as good as the *slack* approach. However, when it uses as its objective function the average cost plus a fixed constant multiplied by the standard deviation of the cost, it generates solutions with less variability.

### 2.2.8 Other nature-inspired heuristics

In Marinakis et al. (2013), the CVRPSD was solved by particle swarm optimization, a methodology based on simulating the social behavior of swarming organisms. The method includes some other heuristics, since in addition to the particle swarm, it also uses path relinking and local search (2-opt and 3-opt). Thus, in some sense it can be viewed as a multi-descent algorithm. In the model of the problem it was assumed that route failures could be avoided by the “optimum choice” of a threshold value to decide whether to travel to the depot for preventive restocking or not. The fitness of a solution is deterministic and evaluated analytically. The results are compared against results obtained by two other evolutionary algorithms, and the particle swarm optimization heuristic was able to outperform them.

In Chepuri and Homem-de Mello (2005) the single vehicle CVRPSD was solved using the Cross Entropy (CE) method (Rubinstein 1999). This method considers that the actual optimization problem is connected to a problem of estimating rare-event probabilities. The idea behind CE is to see the selection of the optimal solution as a rare event. At every iteration a set of routes are generated according to transition probabilities, the cost of the routes is evaluated and probabilities are updated depending on that cost. For the CVRPSD the algorithm starts from an initial transition matrix so any route has the same probability to be generated, and a set of routes is generated. The routes are then evaluated using a demand sample, which is the same for each route. The best route so far is kept and compared to the one generated at each iteration. If no improvement is achieved for a given number of iterations, the algorithm stops. Lower bounds and exact solutions are computed and several type of analysis are done for different cases: demands are IID and penalties are identical; demands are IID and penalties are not identical; demands are non-IID and penalties are identical; demands are non-IID and penalties are non-identical. If demands are IID, the expected cost can be computed analytically. If demands are

non-IID, expected cost of going back to depot is ignored and a lower bound is obtained for the total cost. Demands are drawn from the same family of distributions, but the parameters depend on the node. A lower bound is also obtained for the case where demands are non-IID and the penalties are non-uniform. For IID demands, the algorithm is compared with results of a branch-and-bound technique using the ILOG SOLVER 4.4. In most of the cases the algorithm is able to find solutions within 5% of the optimal solution. The solver is not able to find exact solutions for more than 16 nodes. For non-IID demands and uniform penalties, a lower bound of the problem is found by ILOG SOLVER 4.4. In tighter problems, where the total demand is close to total capacity, the values are closer to the lower bound. For non-IID demands and non-uniform penalties, no lower bound is available, but a similar conclusion is obtained regarding solution quality of tighter problems. In elite sampling, some randomly generated routes are replaced by the best routes found so far, which improves the performance of the algorithm.

Ant colony optimization is used in Woensel et al. (2007) to solve the routing problems with time-dependent travel times. A 2-opt procedure is used together with a mechanism that splits the tours by adding a depot between two customers. This is done until no improvement is achieved or the maximum number of trucks is reached. In addition, the starting time may be shifted if that represents an improvement. The quality of the obtained solutions was compared with the quality of the solutions when just the total distance is minimized. In addition, explicit enumeration is done for small instances to validate the approach. Tests are done using both objective functions: with and without variance of the travel times (except for complete enumeration which does not include the variance in the objective function.) The algorithm was tested on 28 instances from the literature with 32–100 customers in addition to the depot. On average there is a reduction of 22.2% in the travel time. For tests of the second objective function, the roads are randomly selected to have either a high or low coefficient of variation: 50% of roads have high and 50% low. The coefficient of variation of the travel time decreased on average 54.30% with a weight associated to the variance in the interval [0, 0.1]. The cost of that reduction is an increment in the average travel time of 27.87%. In the tests, the fleet size is considered unlimited.

### 3 Tables

A list of all abbreviations and acronyms used in this section can be found in Table 1. In Table 2 there is a summary of the solution methods used in the surveyed papers where customers demand is assumed to be stochastic.

In Table 3 there is a summary of the solution methods used in the surveyed papers where customers and customer demands are assumed to be stochastic.

In Table 4 there is a summary of the solution methods used to deal with the stochastic CARP in the surveyed papers.

In Table 5 there is a summary of the solutions methods used in the surveyed papers where time is assumed to be stochastic.

**Table 1** Notation

Abbreviation/acronym	Method
ACO	Ant colony optimization
ALNS	Adaptive large neighborhood search
B&C	Branch-and-cut
B&P	Branch-and-price
BC&P	Branch-cut-and-price
CCP	Chance constraint programming
CE	Cross entropy
CO	Cyclic-order
CARPSD	Capacitated arc routing problem with stochastic demands
CVRPSCD	Capacitated vehicle routing problem with stochastic customers and demands
CVRPSD	Capacitated vehicle routing problem with stochastic demands
CW	Clarke–Wright
DP	Dynamic programming
DTD	Detour to depot
EA	Evolutionary algorithm
HC	Heuristic concentration
ILS	Iterated local search
LAH	Look ahead heuristic
LBD	Local branching descent
LS	Local search
LSM	L-shaped method
MA	Memetic algorithm
MCGRPC	Mixed capacitated general routing problem with probabilistic constraints
MDLB	Multidescent local branching
MOEA	Multi-objective evolutionary algorithm
MPDP	Multi-vehicle pickup and delivery problem
MSH	Multi-start heuristic
MS-ILS	Multi-start iterated local search
MSSH	Multispace sampling heuristics
MSSP	Multi-start search procedure
NDM	Neuro-dynamic methodology
PHH	Progressive hedging heuristic
PLC	Pair locally coordinated
PSO	Particle swarm optimization
SA	Simulated annealing
SCW	Stochastic Clarke–Wright
SS	Scatter search
TS	Tabu search
VRPSTT	Vehicle routing problem with stochastic travel times

**Table 2** Summary of papers where demands are assumed to be stochastic

Author	Type of problem	Solution method	Evaluation
Hjorring and Holt (1999)	CVRPSD	LSM	AN
Yang et al. (2000)	CVRPSD	DP	AN
Secomandi (2000)	CVRPSD	NDM	AN
Secomandi (2001)	CVRPSD	RA	AN
Laporte et al. (2002)	CVRPSD	LSM	AN
Secomandi (2003)	CVRPSD	RA	AN
Chang (2005)	CVRPSD	LSM	AN
Bianchi et al. (2005)	CVRPSD	ACO, EA, SA, TS and ILS	AN
Chepuri and Homem-de Mello (2005)	CVRPSD	CE	SI
Ak and Erera (2007)	CVRPSD	TS	AN
Christiansen and Lysgaard (2007)	CVRPSD	B&P	AN
Haugland et al. (2007)	CVRPSD	TS, MSH	AN
Rei et al. (2007)	CVRPSD	LBD	AN
Tan et al. (2007)	CVRPSD	MOEA	SI
Sungur et al. (2008)	CVRPSD	B&C	SI
Secomandi and Margot (2009)	CVRPSD	DP	AN
Sørensen and Sevau (2009)	CVRPSD and stochastic travel costs	MA	SI
Novoa and Storer (2009)	CVRPSD	DP	SI
Erera et al. (2010)	CVRPSD	TS	AN
Rei et al. (2010)	CVRPSD	MDLB, LSM	SI
Mendoza et al. (2010)	CVRPSD	MA	AN
Lei et al. (2011)	CVRPSD	ALNS	AN
Mendoza et al. (2011)	CVRPSD	SCW, LAH and DP	AN
Juan et al. (2011)	CVRPSD	MSSP + CW	SI
Goodson et al. (2012)	CVRPSD	SA + CO	AN
Lee et al. (2012)	CVRPSD	B&P	AN
Gounaris et al. (2013)	CVRPSD	B&C	SI
Mendoza and Villegas (2013)	CVRPSD	MSSH	AN
Marinakis et al. (2013)	CVRPSD	PSO	AN
Jabali et al. (2014)	CVRPSD	LSM	AN
Gauvin et al. (2014)	CVRPSD	BC&P	AN
Zhu et al. (2014)	CVRPSD	DP, RA	AN
Goodson (2015)	CVRPSD	SA + CO	AN
Mendoza et al. (2015)	CVRPSD	GRASP + HC	AN

**Table 3** Summary of papers where customers and customer demands are stochastic

Author	Type of problem	Solution method	Evaluation
Hvattum et al. (2006)	CVRPSCD	PHH	SI
Zhong et al. (2007)	CVRPSCD	TS	AN
Erera et al. (2009)	CVRPSCD	LS	SI
Sørensen and Sevaux (2009)	CVRPSCD	MA	SI
Beraldi et al. (2010)	Probabilistic multi-vehicle pickup and delivery problem	LS	AN
Ahmadi-Javid and Seddighi (2013)	Location, allocation and routing problem with disruptions	LS	AN

**Table 4** Summary of papers on stochastic CARP

Author	Type of problem	Solution method	Evaluation
Fleury et al. (2002)	CARPSCD	GA	SI
Fleury et al. (2004)	CARPSCD	GA	AN
Fleury et al. (2005b)	CARPSCD	GA	SI
Fleury et al. (2005a)	CARPSCD	GA	SI
Christiansen et al. (2009)	CARPSCD	B&P	AN
Laporte et al. (2010)	CARPSCD	ALNS	AN
Beraldi et al. (2015)	MCGRPC	B&C, VND	SI
Chen et al. (2014)	CARP with stochastic travel and service times	ALNS	SI

Table 6 shows how often the different methods have been used in the literature. In a single paper more than one method may be used, so the counting here is not done per paper, but per number of times in general. Values are rounded to the closest integer.

## 4 Conclusions

The different methods used in the past 20 years of research on stochastic VRPs have been described and classified. Although both exact and heuristic methods have been used, heuristics are more common. This is probably due to the difficulties inherent in the VRP that are exacerbated by the addition of stochastic elements. Within heuristics, we found that the most common type of methods are classified as local

**Table 5** Summary of papers with time as stochastic parameter

Author	Type of problem	Solution method	Evaluation
Kenyon and Morton (2003)	VRP with stochastic travel and service times	B&C	AN
Ando and Taniguchi (2006)	VRPSTT	GA	SI
Woensel et al. (2007)	VRPSTT	ACO	AN
Russell and Urban (2008)	VRPSTT	TS	AN
Lecluyse et al. (2009)	VRPSTT	TS	AN (aprox)
Li et al. (2010)	VRP with stochastic travel and service times	TS	SI
Lei et al. (2012)	VRP with stochastic service times	GVNS	AN
Lee et al. (2012)	VRPSTT	B&P	AN
Zhang et al. (2012)	VRPSTT	SS	AN
Taş et al. (2013)	VRPSTT	TS	AN
Zhang et al. (2013)	VRP with stochastic travel and service times	TS	AN (aprox)
Taş et al. (2014b)	VRPSTT	B&P	AN
Adulyasak and Jaillet (2016)	VRPSTT	B&C	AN
Taş et al. (2014a)	VRPSTT	TS, ALNS	AN (aprox)
Yan et al. (2014)	Cash transportation under stochastic travel time	Problem decomposition + CPLEX	AN
Ehmke et al. (2015)	VRPSTT	TS	AN (aprox)
Solano-Charris et al. (2015)	VRPSTT	ILS, MS-ILS, CW, GRASP	AN
Gómez et al. (2016)	VRP with stochastic travel and service times	MSSH	SI
Errico et al. (2016)	VRP with stochastic service times	BC&P	AN
Miranda and Conceição (2016)	VRPSTT	ILS	AN (aprox)

search. In this group, tabu search has been the most popular and arguably the most effective.

The SVRP is a computationally demanding problem, so looking to the future we expect to see methods based on simple but effective ideas. In fact, one of the most effective methods for the CVRPSD is based on a relatively simple procedure. A pool of feasible routes is built and feasible solutions are constructed based on such pool.

Another aspect that we expect to become more popular in the SVRP literature is the evaluation of solutions using methods more accurate than simulation but less

**Table 6** Solution methods

Optimization method	Percentage of papers (%)
Local search	36
Tabu search	41
Simulated annealing	14
Adaptive large neighborhood search	14
Iterated local search	10
Other local search algorithms	7
Multidescent local branching	4
Multi-start heuristic	4
Multi-start iterated local search	4
VNS	4
Exact algorithms	25
L-shaped method	35
Branch-and-price	25
Branch-and-cut	25
Branch-cut-and-price	10
Other (problem decomposition + CPLEX)	5
Evolutionary algorithms	11
Genetic algorithm	66
Memetic algorithm	22
Scatter search	11
Constructive algorithms	10
Multispace sampling heuristics	25
GRASP	25
Clarke–Wright	25
Look ahead heuristic	13
Multi-start search procedure + Clarke–Wright	13
Other nature-inspired heuristics	5
Ant colony optimization	50
Particle swarm optimization	25
Cross entropy	25
Other methods	13
Dynamic programming	50
Rollout algorithm	40
Progressive hedging heuristic	10

computationally demanding and that are more flexible than the analytic evaluation by closed forms.

Finally, we expect trends in the stochastic programming literature towards more effective use of parallel computing to find their way into the SVRP literature at an accelerating pace. Although the future is uncertain, we can predict with confidence

that modeling and solving stochastic vehicle routing problems will remain an area of active and important research. After completion of this survey, we became aware of a related survey that provides a dynamic VRP perspective on the literature (Ritzinger et al. 2016).

## References

- Adulyasak Y, Jaillet P (2016) Models and algorithms for stochastic and robust vehicle routing with deadlines. *Transp Sci* 50(2):608–626
- Ahmadi-Javid A, Seddighi AH (2013) A location-routing problem with disruption risk. *Transp Res Part E Log Transp Rev* 53:63–82
- Ak A, Erera AL (2007) A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. *Transp Sci* 41(2):222–237
- Ando N, Taniguchi E (2006) Travel time reliability in vehicle routing and scheduling with time windows. *Netw Spat Econ* 6(3–4):293–311
- Beraldi P, Ghiani G, Musmanno R, Vocaturo F (2010) Efficient neighborhood search for the probabilistic multi-vehicle pickup and delivery problem. *Asia Pac J Oper Res* 27(3):301–314
- Beraldi P, Bruni ME, Lagana D, Musmanno R (2015) The mixed capacitated general routing problem under uncertainty. *Eur J Oper Res* 240(2):382–392
- Bertsimas D, Cervi P, Peterson M (1995) Computational approaches to stochastic vehicle routing problems. *Transp Sci* 29(4):342–352
- Bianchi L, Birattari M, Chiarandini M, Manfrin M, Mastrolilli M, Paquete L, Rossi-Doria O, Schiavinotto T (2005) Hybrid metaheuristics for the vehicle routing problem with stochastic demands. Technical report, Dalle Molle Institute for Artificial Intelligence, Manno
- Chang MS (2005) A vehicle routing problem with time windows and stochastic demands. *J Chin Inst Eng* 28(5):783–794
- Chang TS, Wan YW, Ooi WT (2009) A stochastic dynamic traveling salesman problem with hard time windows. *Eur J Oper Res* 198(3):748–759
- Chen L, Ha MH, Langevin A, Gendreau M (2014) Optimizing road network daily maintenance operations with stochastic service and travel times. *Transp Res Part E Log Transp Rev* 64:88–102
- Chepuri K, Homem-de Mello T (2005) Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Ann Oper Res* 134(1):153–181
- Christiansen CH, Lysgaard J (2007) A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Oper Res Lett* 35(6):773–781
- Christiansen CH, Lysgaard J, Wøhlk S (2009) A branch-and-price algorithm for the capacitated arc routing problem with stochastic demands. *Oper Res Lett* 37(6):392–398
- Clarke C, Wright J (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res* 12:432–441
- Cordeau J-F, Gendreau M, Laporte G, Potvin J-Y, Semet F (2002) A guide to vehicle routing heuristics. *J Oper Res Soc* 53(5):512
- Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Manag Sci* (pre-1986) 6(1):80
- Desrochers M (1988) An algorithm for the shortest path problem with resource constraints. In: Technical report, Cahiers du GERAD G-88-27, University of Montreal
- Dueck G (1993) New optimization heuristics: the Great Deluge algorithm and the record-to-record travel. *J Comput Phys* 104(1):86–92
- Ehmke JF, Campbell AM, Urban TL (2015) Ensuring service levels in routing problems with time windows and stochastic travel times. *Eur J Oper Res* 240(2):539–550
- Eksioglu B, Vural AV, Reisman A (2009) The vehicle routing problem: a taxonomic review. *Comput Ind Eng* 57(4):1472–1483
- Erera AL, Savelsbergh M, Uyar E (2009) Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. *Networks* 54(4):270–283
- Erera AL, Morales JC, Savelsbergh M (2010) The vehicle routing problem with stochastic demand and duration constraints. *Transp Sci* 44(4):474–492

- Errico F, Desaulniers G, Gendreau M, Rei W, Rousseau L-M (2016) A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *Eur J Oper Res* 249(1):55–66
- Fallahi AE, Prins C, Wolfier Calvo R (2008) A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Comput Oper Res* 35(5):1725–1741
- Feillet D, Dejax P, Gendreau M, Gueguen C (2004) An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. *Networks* 44(3):216–229
- Fleury G, Lacomme P, Prins C, Randame-Chérif W (2002) Robustness evaluation of solutions for the capacitated arc routing problem. In: Barros FJ, Gambiasi N (eds) *Simulation and planning in high autonomy systems*. CARP, Lisbon
- Fleury G, Lacomme P, Prins C (2004) Evolutionary algorithms for stochastic arc routing problems. *Lecture notes in computer science*, vol 3005. Springer, Berlin, pp 501–512
- Fleury G, Lacomme P, Prins C, Ramdane-Cérief W (2005) Improving robustness of solutions to arc routing problems. *J Oper Res Soc* 56(5):526–538
- Fleury G, Philippe L, Prins C (2005) Stochastic capacitated arc routing problem. In: Technical report LIMOS/RR-05-12, Laboratoire d'informatique (LIMOS), Université Blaise Pascal
- Gauvin C, Desaulniers G, Gendreau M (2014) A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Comput Oper Res* 50:141–153
- Gendreau M, Laporte G, Seguin R (1996) A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Oper Res* 44(3):469–477
- Gendreau M, Jabali O, Rei W (2014) Stochastic vehicle routing problems. In: Toth P, Vigo D (eds) *Vehicle routing: problems, methods, and applications*, 2nd edn, Society for Industrial and Applied Mathematics, pp 213–239
- Gómez A, No RM, Akhavan-Tabatabaei R, Medaglia AL, Mendoza JE (2016) On modeling stochastic travel and service times in vehicle routing. *Transp Sci* 50(2):627–641
- Goodson JC (2015) A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands. *Eur J Oper Res* 241(2):361–369
- Goodson JC, Ohlmann JW, Thomas BW (2012) Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. *Eur J Oper Res* 217(2):312–323
- Gounaris CE, Wiesemann W, Floudas CA (2013) The robust capacitated vehicle routing problem under demand uncertainty. *Oper Res* 61(3):677–693
- Groër C, Golden B, Wasil E (2009) The consistent vehicle routing problem. *Manuf Serv Oper Manag* 11(4):630–643
- Haugland D, Ho SC, Laporte G (2007) Designing delivery districts for the vehicle routing problem with stochastic demands. *Eur J Oper Res* 180(3):997–1010
- Hjorring C, Holt J (1999) New optimality cuts for a single-vehicle stochastic routing problem. *Ann Oper Res* 86:569–584
- Hvattum LM, Løkketangen A, Laporte G (2006) Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transp Sci* 40(4):421–438
- Jabali O, Rei W, Gendreau M, Laporte G (2014) Partial-route inequalities for the multi-vehicle routing problem with stochastic demands. *Discrete Appl Math* 177:121–136
- Juan A, Faulin J, Grasman S, Riera D, Marull J, Mendez C (2011) Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. *Transp Res Part C Emerg Technol* 19(5):751–765 (Freight Transportation and Logistics (selected papers from ODYSSEUS 2009—the 4th international workshop on Freight Transportation and Logistics))
- Kenyon AS, Morton DP (2003) Stochastic vehicle routing with random travel times. *Transp Sci* 37(1):69–82
- Lacomme P, Prins C, Ramdane-Chérif W (2001) A genetic algorithm for the capacitated arc routing problem and its extensions. In: Boers E (ed) *Lecture notes in computer science*, vol 2037. Springer, Berlin, pp 473–483
- Laporte G, Louveaux FV, Hamme LV (2002) An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Oper Res* 50(3):415–423
- Laporte G, Musmanno R, Vocaturo F (2010) An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transp Sci* 44(1):125–135
- Lecluyse C, Van Woensel T, Peremans H (2009) Vehicle routing with stochastic time-dependent travel times. *4OR* 7(4):363–377

- Lee C, Lee K, Park S (2012) Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *J Oper Res Soc* 63(9):1294–1306
- Lei H, Laporte G, Guo B (2011) The capacitated vehicle routing problem with stochastic demands and time windows. *Comput Oper Res* 38(12):1775–1783
- Lei H, Laporte G, Guo B (2012) A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times. *TOP* 20(3):99–118
- Li X, Tian P, Leung SC (2010) Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. *Int J Prod Econ* 125(1):137–145
- Maden W, Eglese R, Black D (2010) Vehicle routing and scheduling with time-varying data: a case study. *J Oper Res Soc* 61(3):515–522
- Marinakis Y, Iordanidou G-R, Marinaki M (2013) Particle swarm optimization for the vehicle routing problem with stochastic demands. *Appl Soft Comput* 13(4):1693–1704
- Mendoza J, Villegas J (2013) A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optim Lett* 7(7):1503–1516
- Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2010) A memetic algorithm for the multicompartment vehicle routing problem with stochastic demands. *Comput Oper Res* 37(11):1886–1898
- Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2011) Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transp Sci* 45(3):346–363
- Mendoza J, Rousseau L-M, Villegas J (2015) A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints. *J Heuristics* 1–28
- Miranda DM, Conceição SV (2016) The vehicle routing problem with hard time windows and stochastic travel and service time. *Expert Syst Appl* 64:104–116
- Novoa C, Storer R (2009) An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *Eur J Oper Res* 196(2):509–515
- Oyola J, Arntzen H, Woodruff DL (2016) The stochastic vehicle routing problem, a literature review, part I: models. *EURO J Transp Logist.* doi:[10.1007/s13676-016-0100-5](https://doi.org/10.1007/s13676-016-0100-5)
- Prins C (2004) A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput Oper Res* 31(12):1985–2002
- Rei W, Gendreau M, Soriano P (2007) Local branching cuts for the 0–1 integer L-shaped algorithm. In: Technical report, CIRRELT-2007-23, Centre interuniversitaire de recherche sur les réseaux d’entreprise, la logistique et le transport, vol 44(1), pp 136–146
- Rei W, Gendreau M, Soriano P (2010) A hybrid Monte Carlo local branching algorithm for the single vehicle routing problem with stochastic demands. *Transp Sci* 44(1):136–146
- Resende MGC, Celso CR (2010) Greedy randomized adaptive search procedures: advances, hybridizations, and applications. In: Gendreau M, Potvin J-Y (eds) *Handbook of metaheuristics*. Springer, New York, pp 283–319
- Ritzinger U, Puchinger J, Hartl RF (2016) A survey on dynamic and stochastic vehicle routing problems. *Int J Prod Res* 54:215–231
- Rockafellar R, Wets R-B (1991) Scenarios and policy aggregation in optimization under uncertainty. *Math Oper Res* 16(1):119–147
- Rubinstein R (1999) The cross-entropy method for combinatorial and continuous optimization. *Methodol Comput Appl Probab* 1(2):127–190
- Russell RA, Urban TL (2008) Vehicle routing with soft time windows and erlang travel times. *J Oper Res Soc* 59(9):1220–1228
- Secomandi N (2000) Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Comput Oper Res* 27(1112):1201–1225
- Secomandi N (2001) A rollout policy for the vehicle routing problem with stochastic demands. *Oper Res* 49(5):796–802
- Secomandi N (2003) Analysis of a rollout approach to sequencing problems with stochastic routing applications. *J Heuristics* 9(4):321–352
- Secomandi N, Margot F (2009) Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Oper Res* 57(1):214–230
- Solano-Charris E, Prins C, Santos AC (2015) Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios. *Appl Soft Comput* 32:518–531
- Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper Res* 35(2):254

- Sörensen K, Sevaux M (2009) A practical approach for robust and flexible vehicle routing using metaheuristics and Monte Carlo sampling. *J Math Model Algorithms* 8(4):387–407
- Sungur I, Ordóñez F, Dessouky M (2008) A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Trans* 40:509–523
- Sungur I, Ren Y, Ordez F, Dessouky M, Zhong H (2010) A model and algorithm for the courier delivery problem with uncertainty. *Transp Sci* 44(2):193–205
- Taş D, Dellaert N, van Woensel T, de Kok T (2013) Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Comput Oper Res* 40(1):214–224
- Taş D, Dellaert N, van Woensel T, de Kok T (2014a) The time-dependent vehicle routing problem with soft time windows and stochastic travel times. *Transp Res Part C Emerg Technol* 48:66–83
- Taş D, Gendreau M, Dellaert N, van Woensel T, de Kok A (2014b) Vehicle routing with soft time windows and stochastic travel times: a column generation and branch-and-price solution approach. *Eur J Oper Res* 236(3):789–799
- Tan K, Cheong C, Goh C (2007) Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *Eur J Oper Res* 177(2):813–839
- Toth P, Vigo D (1998) Exact algorithms for vehicle routing. In: Crainic T, Laporte G (eds) *Fleet management and logistics*. Kluwer Academic Publishers, New York, pp 1–31
- Toth P, Vigo D (2002) Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Appl Math* 123(1–3):487–512
- Voß S, Fink A, Duin C (2005) Looking ahead with the pilot method. *Ann Oper Res* 136(1):285–302
- Woensel TV, Kerbache L, Peremans H, Vandaele N (2007) A queueing framework for routing problems with time-dependent travel times. *J Math Model Algorithms* 6(1):151–173
- Yan S, Wang S-S, Chang Y-H (2014) Cash transportation vehicle routing and scheduling under stochastic travel times. *Eng Optim* 46(3):289–307
- Yang W-H, Mathur K, Ballou RH (2000) Stochastic vehicle routing problem with restocking. *Transp Sci* 34(1):99–112
- Zhang T, Chaovallitwongse W, Zhang Y (2012) Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Comput Oper Res* 39(10):2277–2290
- Zhang J, Lam WHK, Chen BY (2013) A stochastic vehicle routing problem with travel time uncertainty: trade-off between cost and customer service. *Netw Spat Econ* 13(4):471–496
- Zhong H, Hall RW, Dessouky M (2007) Territory planning and vehicle dispatching with driver learning. *Transp Sci* 41(1):74–89
- Zhu L, Rousseau L-M, Rei W, Li B (2014) Paired cooperative reoptimization strategy for the vehicle routing problem with stochastic demands. *Comput Oper Res* 50:1–13