

A hybrid metaheuristic to solve the vehicle routing problem with stochastic demand and probabilistic duration constraints

Jorge E. Mendoza

LUNAM Université - LISA (EA 4094 CNRS)
Université Catholique de l'Ouest, Angers, France
Email: jorge.mendoza@uco.fr

Louis-Martin Rousseau

CIRRELT, École Polytechnique de Montréal, Canada
Email: louis-martin.rousseau@cirrelt.ca

Juan G. Villegas

Departamento de Ingeniería Industrial
Universidad de Antioquia, Medellín, Colombia
Email: jvillega@udea.edu.co

1 Introduction

The vehicle routing problem with stochastic demands and probabilistic distance constraints (VRPSD-PDC) can be defined on a complete and undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{0, \dots, n\}$ is the vertex set and $\mathcal{E} = \{(v, u) : v, u \in \mathcal{V}, v \neq u\}$ is the edge set. Vertices $v = 1, \dots, n$ represent the customers and vertex $v = 0$ represents the depot. A distance d_e is associated with edge $e = (v, u) = (u, v) \in \mathcal{E}$, and it represents the travel cost between vertices v and u . Each customer v has a random demand ξ_v for a given product. Customer demands are met using an unlimited fleet of homogeneous vehicles located at the depot. Each vehicle has a maximum capacity Q and a maximum travel distance L . The exact quantity demanded by each customer is not known until the vehicle arrives at the customer location. It is assumed, however, that each customer's demand follows an independent and known probability distribution and that all demand realizations (actual quantities) are nonnegative and less than the capacity of the vehicle.

We formulate the VRPSD-PDC as an extension of the classical two-stage stochastic programming formulation for the VRPSD that includes chance constraints on the maximum travel distances of the routes. As the name suggests, the two-stage stochastic

programming formulation for the VRPSD solves the problem in two stages. In the first stage, a set \mathcal{R} of *planned* routes is designed. Each route $r \in \mathcal{R}$ is a sequence of vertices $r = (0, v_1, \dots, v_i, \dots, v_{n_r}, 0)$, where $v_i \in \mathcal{V} \setminus \{0\}$ and n_r is the number of customers serviced by the route. During the planning phase, each route is designed so that the total expected load does not exceed the capacity of the vehicle (i.e., $\sum_{v \in r \setminus 0} E[\xi_v] \leq Q \forall r \in \mathcal{R}$) and every customer is visited by exactly one route. In the second stage, each route is executed until a *route failure* occurs, that is, the capacity of the vehicle is exceeded. A *recourse action* is then applied to recover the feasibility of the failing route. The recourse action is classically defined as a return to the depot to reload the vehicle, followed by a trip back to the customer location to complete the service. The route is then resumed from that point as originally planned. It is worth noting that the literature includes more sophisticated recourse actions. We decided to retain the classical policy because it is simple, suitable for many practical applications, and allows a more direct comparison with previously published results. The second-stage solution is then the actual set of routes traveled by the vehicles. The problem is to determine in the first stage the set of planned routes \mathcal{R} that minimizes the expected cost $E[C]$ of the second-stage solution given by:

$$E[C(\mathcal{R})] = \sum_{r \in \mathcal{R}} E[C_r] = \sum_{r \in \mathcal{R}} E[l_r + G_r(\vec{\xi})] = \sum_{r \in \mathcal{R}} l_r + \sum_{r \in \mathcal{R}} E[G_r(\vec{\xi})] \quad (1)$$

where l_r is the planned length (planned cost) and $E[G_r(\vec{\xi})]$ is the expected length of the return trips to the depot, or the cost of recourse, caused by route failures for each route $r \in \mathcal{R}$. The planned cost of a route is simply the sum of the lengths of the arcs traversed by the route. Under the selected recourse action, the expected cost of the failures of a route is given by $E[G_r(\vec{\xi})] = 2 \times \sum_{i=2}^{n_r} Pr(v_i) \times d_{v_i,0}$, where $Pr(v_i)$ represents the probability of having a route failure while servicing customer v_i . The *failure probability* is calculated as $Pr(v_i) = \sum_{f=1}^{i-1} Pr(\sum_{j=2}^{i-1} \xi_{v_j} \leq f \cdot Q < \sum_{j=2}^i \xi_{v_j})$, where the probability term represents the probability of having the f_{th} failure while servicing customer v_i .

Since the total travel distance of each route (i.e., C_r) is a random variable which value is only known when the vehicle returns to the depot after completing the route. The literature accounts for different strategies to deal with the distance constraint in the context of the VRPSD. For instance, Yang et al. (2000) enforce the distance constraint on the total expected cost of the planned route; Erera et al. (2010) impose a hard constraint on the second-stage cost of the route, meaning that planned routes should verify the constraint for any possible vector of demand realizations; and Tan et al. (2007) penalize violations to the distance constraint in an additional objective function, driver remuneration, and solve the problem as a multiobjective optimization problem with posterior articulation of preferences. Alternatively, we model the distance constraint stating that:

$$Pr(C_r \leq L) \geq 1 - \beta \quad \forall r \in \mathcal{R} \quad (2)$$

where β is an acceptance threshold set by the decision maker according to his risk aversion to violations to the distance constraint.

In this research, we propose an algorithm to analytically compute $Pr(C_r \leq L)$, and thus evaluate constraints (2), and use that algorithm as a building block to construct a GRASP with heuristic concentration (HC) for solving the VRPSD-PDC.

2 GRASP with heuristic concentration

The proposed approach operates as follows. At each iteration t the algorithm selects a randomized TSP heuristic $h \in H$ and uses it to build a giant TSP tour p^t visiting all customers. Then, the algorithm uses an adaptation of the the s-split procedure for the VRPSD (Mendoza and Villegas, 2012) to optimally partition p^t into a set of feasible routes that make up a starting solution s^t . Next, the algorithm uses the first-improvement versions of the re-locate and 2-opt neighborhoods to perform a Variable Neighborhood Descent (VND) from the starting solution s^t . At the end of iteration t , our GRASP updates the best known solution s^* and adds the routes in the local optimum (i.e., s^t) to a set Ω . After T iterations the GRASP stops and the heuristic concentration takes place. In this phase, our method uses a commercial optimizer to solve a set-partitioning formulation for the VRPSD-PDC over the set of routes Ω . To speed up the HC phase, the algorithm uses the objective function of the best solution found by the GRASP (i.e., $f(s^*)$) as an initial upper bound for the set-partitioning problem.

Note that computing both the objective function (1) and the distance constraints (2) while testing every single move in the local search phase may be computationally prohibiting. To overcome this difficulty, our VND evaluates moves following a four-stage hierarchical procedure. Let s be a search solution and s' be the solution resulting from applying a given move to s . The move evaluation procedure operates as follows. In the first stage the algorithm checks that s' verifies the expected load constraint. In the second stage the algorithm calculates the planned cost of the candidate solution $l_{s'}$ and tests the following condition: $l_{s'} \leq E[C(s)] + \alpha \times E[C(s)]$, where $\alpha \in [-1, 1]$ is a pre-tuned parameter; the procedure moves to the third stage only if the condition is verified. The intuition behind this filter comes from the fact that the planned cost of a solution tends to largely dominate the cost of recourse. Therefore, by setting α to close-to-zero values, the algorithm is able to save expensive computations of the cost of recourse by rapidly discarding moves that are unlikely to improve the solution (at the risk of rejecting some improving moves). In the third stage, the algorithm completes the evaluation of the objective function by computing the cost of recourse of s' . The evaluation procedure advances to the fourth stage only if s' is an improving solution. On the fourth stage the algorithm verifies the distance constraint (2) and then accepts or rejects the move.

3 Computational experiments

We ran 10 times our GRASP+HC with $\beta = 1$ on each instance of the 40-instance testbed for the classical VRPSD introduced in Christiansen and Lysgaard (2007) and compared our results with the best known solutions for the testbed: 38 optimal solutions reported in Gauvin (2012) and 2 heuristic solutions reported in Mendoza and Villegas (2012). We obtained solutions with a maximum gap with respect to the BKSs of 0.03%. From each of the 40 VRPSD instances, we built 2 VRPSDPDC instances by setting L to $\max\{E[C_r]|r \in \mathcal{R}\}$ and $1.2 \times \max\{E[C_r]|r \in \mathcal{R}\}$, where \mathcal{R} is the best solution found for the original VRPSD instance. To analyze the impact of the probabilistic distance constraint on the cost of VRPSD solutions, we ran our algorithm with $\beta = \{0.05, 0.10\}$ on each of the 80 new VRPSDPDC instances. The data show that our method is able to generate solutions that are more robust to violation of the distance constraint than those obtained for the original VRPSD instances with small increments on the expected cost.

References

- C. Christiansen and J. Lysgaard. A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters*, 35(6):773–781, 2007.
- A. Erera, J. C. Morales, and M. Savelsbergh. The vehicle routing problem with stochastic demands and duration constraints. *Transportation Science*, 44(4):474–492, 2010.
- C. Gauvin. Un algorithme de génération de colonnes pour le problème de tournées de véhicules avec demandes stochastiques. Master’s thesis, École Polytechnique de Montréal, 2012.
- J. E. Mendoza and J. G. Villegas. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optimization Letters*, 2012. DOI 10.1007/s11590-012-0555-8.
- K. C. Tan, C. Y. Cheong, and C. K. Goh. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, 177(2):813–839, 2007.
- W. H. Yang, K. Mathur, and R. Ballou. Stochastic vehicle routing with restocking. *Transportation Science*, 34(1):99–112, 2000.