A space biased-sampling approach for the vehicle routing problem with stochastic demands

Jorge E. Mendoza, Juan Villegas

A space biased-sampling approach for the vehicle routing problem with stochastic demands

Jorge E. Mendoza[1], Juan G. Villegas[2]

[1] Équipe Optimisation des Systèmes de Production et Logistiques, LISA (EA CNRS 4094), Université Catholique de l'Ouest. 3 Place André Leroy 49008 Angers, France
jorge.mendoza@uco.fr

[2] Departamento de Ingeniería Industrial, Universidad de Antioquia Calle 70 No. 52 - 21 Medellin, Colombia
jvillega@udea.edu.co

Abstract

The vehicle routing problem with stochastic demands consists of designing transportation routes of minimal expected cost to satisfy a set of customers with random demands of known probability distribution. We propose a metaheuristic that uses randomized heuristics for the traveling salesman problem, a tour partitioning procedure, and a set-partitioning formulation to sample the solution space and find solutions for the problem. Computational experiments show that our approach is competitive with state-of-the-art algorithms for the problem in terms of both accuracy and efficiency.

1 Introduction

In the vehicle routing problem with stochastic demands (VRPSD) a set of geographically-spread customers demand (or supply) a product that must be delivered (or picked-up) using a fleet of homogeneous vehicles located at a central depot. Each vehicle has a fixed and limited capacity, but the number of vehicles that can be used to service the demands is unlimited. The particular characteristic of the problem is that the exact quantities demanded (supplied) by each customer are only known upon the vehicle's arrival to the customer location (i.e., are stochastic). It is assumed, however, that each customer's demand follows an independent and known probability distribution. It is also assumed that the demand realizations (actual quantities) are nonnegative and less than the capacity of the vehicle.

The main effect of the stochastic demands is that if their realizations turn out to be larger than expected, the vehicle capacity may be exceeded leading to what is commonly known as a *route failure*. Different solution frameworks can be applied to deal with the uncertain demands in the VRPSD [4]. Among these frameworks, the most widely used in the literature, and the one we selected for this research, is two-stage stochastic programming. As it name suggests, this framework solves the VRPSD in two stages. In the first stage, a set of *planned* routes is designed. Each route is a tour starting and ending at the depot after visiting a set of customers. In the first-stage solution, each customer must be visited by exactly one vehicle (route) and the total expected demand serviced by each vehicle cannot exceed its capacity. In the second stage, each planned route is executed until a route failure occurs. Upon failure, a recourse action is applied to recover the feasibility of the failing route. The recourse action is classically defined as a return trip to the depot to restore the capacity of the vehicle, followed by a trip back to the customer location to complete the service. After service completion, the route is resumed from that point on as originally planned. The second stage solution is then the actual set of routes traveled by the vehicles. It is worth noting that the literature accounts for more sophisticated recourse actions; nonetheless, we decided to keep the classical policy because: it is simple, suitable to many practical applications, and it allows a more direct comparison with previously published results. Solving the two-stage stochastic program for the VRPSD consists in selecting in the first stage the set of planned routes that minimizes the expected cost of the second stage solution. For details on the computation of the expected cost of the second stage solution the reader is referred to [1, 3, 4].

Solution approaches for the VRPSD based on the two-stage stochastic programming formulation described above include both exact [1, 3] and heuristic [2, 4] methods. As it is often the case in combinatorial optimization problems, exact methods for the VRPSD are limited to solve small-size instances,

thus industrial-scale problems should be tackled using heuristic approaches. To the best of our knowledge, the two most recently-published heuristic approaches for the VRPSD are the look-ahead heuristics by Mendoza et al. [4] and the simulated annealing by Goodson [2]. The former, are a set of fast, simple, and effective constructive procedures; while the latter is a performing metaheuristic that embeds sophisticated neighborhood schemes. This paper proposes a new heuristic approach that uses randomized heuristics for the traveling salesman problem (TSP), a route partitioning procedure, and a set-partitioning model, to sample the solution space and find high-quality solutions for the problem.

2 Space biased-sampling

When solving the VRPSD (and VRPs in general) one often works with elements that belong to four sets: i) the set of solutions, $\mathcal{S}$; ii) the set of feasible routes, $\mathcal{R}$; iii) the set of feasible clusters of customers, $\mathcal{C}$; and iv) the set of TSP-like tours (i.e., giant tours visiting all customers), $\mathcal{P}$. Heuristic approaches generally rely on searching elements in one set and then mapping them to an element in the set of solutions. For instance, cluster-first, route-second approaches consist in using some procedure to select a sub-set of elements from $\mathcal{C}$ and then apply another procedure to map the selected elements to a single element in set $\mathcal{S}$. More elaborate approaches map elements from different sets iteratively. One good example is Prins' well-known evolutionary algorithm for the capacitated VRP [5]. At each iteration, the algorithm selects two elements from $\mathcal{P}$ and applies to them a crossover operator to obtain a new element. The new element is mapped to an element in $\mathcal{S}$ using the split procedure. Then, local search is applied to the mapped element trying to identify a new (better) element. When local search is completed, the new element is mapped back to $\mathcal{P}$ using a concatenation procedure and the whole routine starts over. In both cases, one-time or iteratively mapping, the effectiveness of the methods is highly dependent on the procedures used to select elements from the sets and the procedures used to map elements between sets.

The metaheuristic proposed in this paper works in two phases. In the first phase, it samples T elements from the set of TSP-like tours (i.e., $\mathcal{P}$) and maps each sampled element p to a sub-set $\mathcal{W}_p$ in the set of feasible routes (i.e., $\mathcal{R}$). In the second phase, the approach maps a set $\mathcal{W} \subset \mathcal{R}$, where $\mathcal{W} = \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_T$, to one element $s \in \mathcal{S}$ by solving a set-partitioning formulation of the problem.

To sample elements from $\mathcal{P}$, our approach uses a set of 4 randomized TSP heuristics, namely, nearest neighbor (NN), nearest insertion (NI), best insertion (BI), and farthest insertion (FI). The four heuristics follow the same principle: at each iteration they grow a tour by selecting a non-visited customer and connecting it to the circuit. Each heuristic, however, uses a different decision rule to select and connect customers. To produce randomized versions of the four heuristics we simply allow each procedure to randomly select the connecting customer between the K top-ranked customers according to its original decision rule. Since there exist TSP-like tours that our randomized heuristics cannot build, we say that our metaheuristic does a biased sampling of set $\mathcal{P}$.

To map each sampled element p to its corresponding $\mathcal{W}_p \subset \mathcal{R}$ our approach uses the s-split procedure for the VRPSD [4]. S-split was originally proposed to map an element in $\mathcal{P}$ to an element in $\mathcal{S}$ by optimally partitioning the giant tour into a set of feasible routes that form a VRPSD solution. Nonetheless, s-split accomplishes its mission by running a dynamic programming algorithm that evaluates every single feasible route that can be obtained from the giant tour without altering the order of the customers. In other words, by saving a reference to the set of routes evaluated by s-split while partitioning p, we obtain a mapping of p to a sub-set $\mathcal{W}_p \subset \mathcal{R}$. Since we also obtain a mapping of p to a solution $s \in \mathcal{S}$, we keep a reference to the best solution observed during the sampling phase to have a good upper bound in the second phase of our metaheuristic.

In the second phase, our approach maps the set $\mathcal{W} = \mathcal{W}_1 \cup \ldots \cup \mathcal{W}_T$ to a solution $s \in \mathcal{S}$ by solving a set-partitioning formulation (model) of the VRPSD. In the model we: define one binary decision variable for each feasible route in $\mathcal{W}$; set the coefficient of each variable in the objective function as the expected cost of the associated route; and set constraints guaranteeing that each customer will be visited by exactly one route. The objective is then to select the best sub-set of routes from $\mathcal{W}$ to build the solution reported by our approach. It is well known that VRPs can be modeled as set-covering/set-partition problems, but

researchers often underestimate the benefits of embedding this approach within metaheuristics. Recently, authors like Villegas [6] have shown that applied as a post-optimization procedure, a set-partitioning model may lead to significant improvements in the quality of final solutions. In the proposed approach, the set-partitioning model is an integral part of the method rather than only a post-optimization strategy.

3 Computational experiments

We implemented our approach in Java (jre V.1.6.0_22-b04) and used the Gurobi Optimizer (version 4.0.1) for solving the set-partitioning model. To test the effectiveness of our approach, we ran it on the 40-instance testbed proposed by Christiansen and Lysgaard [1]. The instances range from 16 to 60 customers and assume Poisson distributed customer demands. For each instance we compared our solution with the best known solution (BKS) which is due to either [1], [2], [4], or to the three of them. It is worth mentioning that [1] provided optimality certificates for 19 out of the 40 instances. For the experiment, we set parameter T to $6,000$ and draw $\frac{T}{4}$ samples using each randomized heuristic. The parameter K of the heuristics was set to 6 for NI, BI, FI, and to 3 for NN.

The results of our experiment show that the proposed approach is effective to solve the VRPSD. On the instances with proven optimum, our method matched the optimal solution on 10 out of 19 instances and delivered solutions with optimality gaps ranging from 0.02% to 0.92% (with an average of 0.50%) on the remaining 9. On the 21 instances without provably optimal solution, our method unveiled 2 new best known solutions and matched another 4. On the remaining 15 instances, our approach provided solutions with gaps (with respect to BKSs) ranging from 0.01% to 1.88% (with an average of 0.56%). These are remarkable results taking into account the absence of a local search procedure, a component that is often essential to build competitive VRP metaheuristics. The experiment also shows that our approach is competitive in terms of computational efficiency. Our method reported execution times ranging from 4 seconds to 161 seconds[1], with an average running time of 47.37 seconds. To illustrate, on the same testbed Goodson reported for his simulated annealing [2], implemented in C++, minimum and maximum execution times of 9s and 603s, respectively (average 268s)[2]. In conclusion the proposed approach is competitive with state-of-the-art algorithms for the VRPSD in terms of both quality and efficiency.

Research currently underway includes the implementation of new randomized heuristics to foster more diversification and local search procedures to add some intensification to the sampling phase.

References

[1] 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.

[2] J. Goodson. *Solution methodologies for vehicle routing problems with stochastic demand*. PhD thesis, University of Iowa, 2010.

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

[4] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N Velasco. Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transportation Science*, In Press. DOI 10.1287/trsc.1100.0353.

[5] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[6] J. G. Villegas. *Vehicle routing problems with trailers*. PhD thesis, Université de Technologie de Troyes (France) / Universidad de los Andes (Colombia), 2010.

[1] On a 3.0GHz Intel Pentium Core Duo processor with 3GB of RAM, running Windows XP professional.

[2] On a 2.8GHz Intel Xeon X5660 processor with 12GB of RAM, running the CentOS 5.3 operating system.