

A genetic algorithm for the multi-compartment vehicle routing problem with stochastic demands

J. Mendoza^{*†} B. Castanier^{*} C. Guéret^{*} A. L. Medaglia[†] N. Velasco[†]

^{*}IRCCyN (UMR CNRS 6597), École des Mines de Nantes
B.P. 20722 F-44307 Nantes Cedex 1, France
{jorge-ernesto.mendoza, bruno.castanier, christelle.gueret}@emn.fr

[†]Departamento de Ingeniería Industrial, Universidad de los Andes
Cr 1E No. 19A-10 Bogotá, Colombia
{jor-mend, amedagli, nvelasco}@uniandes.edu.co

1 Introduction

Classical Vehicle Routing Problems (VRPs) normally assume perfect knowledge of the problem parameters such as clients' demands, and traveling and service times. However, in real world scenarios, very often this is a strong assumption, and the stochastic nature of such parameters should be addressed by specially targeted algorithms [2]. To tackle this aspect, a great research effort has been devoted to study the so called stochastic VRPs (SVRPs).

One of the most studied SVRPs is the VRP with stochastic demands (VRPSD) [4]. The problem consists of finding the optimal set of feasible routes to service a set of clients whose demand is a random variable of known distribution. The VRPSD is recognized to be computationally intractable [4] and large instances can only be handled by heuristic approaches [2]. A generalization of the VRPSD is the multi-compartment VRP with stochastic demands (MC-VRPSD), a problem where clients have stochastic demands for several products, but due to incompatibility constraints they must be transported on separate compartments.

In practice, the MC-VRPSD naturally arises in the dairy industry. Daily, milk from different breeds of dairy cattle are to be collected from a large number of farms located on the country side. Since each type of milk is an ingredient for different products, vehicles with multiple compartments prevent different milk types from mixing. The uncertainty on the demand comes from random factors that affect the productivity of the cattle and from the best economic strategy of the producers.

The interest in solving real world multi-compartment VRPs (MC-VRPs) has increased in recent years. Avella et al. [1] solved a distribution problem where different types of fuel are delivered in tankers with multiple compartments. Oppen and Løkkentangen [7] targeted a livestock transportation problem, where different animals are transported from ranches to slaughterhouses on independent compartments to avoid stress that could result on poor quality

meat. More recently, El Fallahi et al. [3] tackled a variant of MC-VRP where each product can be delivered to the clients by a different vehicle although the full delivery of a single product must be performed by only one vehicle. The cited cases, however, are solved only in deterministic scenarios.

Despite the practical applicability of the MC-VRPSD, to the best of our knowledge the only reference to the problem is the set of construction heuristics proposed in [6]. To fill this void, this paper presents a genetic algorithm (GA) designed to solve a MC-VRPSD. The algorithm couples components proven to be effective on deterministic VRPs with two different evaluation strategies designed to guide the search considering the stochastic nature of the problem. To analyze the performance of both strategies, the algorithm was tested on a large set of instances.

2 Problem definition

Formally, the MC-VRPSD can be defined on a complete and undirected graph $G = (\mathcal{N}, E)$ where $\mathcal{N} = \{0, \dots, n\}$ is the node set and E the edge set. Nodes $i = 1 \dots n$ represent the clients and node $i = 0$ represents the depot. A distance d_e is associated to edge $e = (i, j) = (j, i) \in E$ and it represents the travel cost between nodes i and j . It is assumed that the triangular inequality is satisfied (i.e., $d_{(i,j)} + d_{(j,k)} \geq d_{(i,k)}$). There exists a set $\mathcal{P} = \{1, \dots, p, \dots, m\}$ of products that must be transported on independent compartments. Hence, each vehicle has a dedicated compartment of fixed capacity Q_p for each product. All vehicles are identical and the fleet size unrestricted. For a product p the client i has an independent random demand $\xi_{i,p}$ following a normal distribution $\mathcal{N}(\mu_{i,p}, \sigma_{i,p})$. The actual values of the demands (realizations) are only known upon arrival to the client's location and all realizations are nonnegative and less than the capacity of the corresponding compartment Q_p . Finally, each node must be visited by exactly one route and the total length of each route l_r cannot exceed a maximum distance L .

The MC-VRPSD can be formulated as a two-stage stochastic programming model. On the first stage, a set \mathcal{R} of a priori routes is selected. Each route $r \in \mathcal{R}$ is a sequence of nodes $r = (0, i_1, \dots, i_k, \dots, i_{n_r}, 0)$, where $i_k \in \mathcal{N} \setminus \{0\}$ and n_r is the number of client nodes in route r . On the second stage, each planned route is executed until a route failure occurs. A *route failure* is said to occur whenever the capacity of one compartment is exceeded while visiting a client. When failing, the compartment is loaded up to its capacity and a *recourse* action takes place. The recourse action is defined as a return trip to the depot to unload the compartments, followed by a trip back to the client's location to complete the service. After service completion, the route is resumed from that point on as originally planned. The problem is then to determine the set of routes \mathcal{R} that minimizes the total expected cost $E[C]$ given by:

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

where the first term is the planned length of the routes in \mathcal{R} and $E[G_r(\mathcal{R}, \xi)]$ is the expected length of the return trips caused by route failures.

3 Genetic Algorithm

3.1 General structure

The proposed GA encodes the MC-VRPSD solutions into a multipermutation genotype known as the Genetic Vehicle Representation (GVR) [8]. Specifically, each permutation contains an ordered set of clients representing a route. Starting from an initial population $\mathcal{P}(0)$ comprised of P individuals, the algorithm follows an evolutionary process of T generations. At every generation t , crossover and mutation operators are applied with probabilities p_c and p_m on each individual of population $\mathcal{P}(t)$. The offspring produced by the genetic operators form a new children population $\mathcal{C}(t) = \mathcal{C}_c(t) \cup \mathcal{C}_m(t)$. Both $\mathcal{C}(t)$ and $\mathcal{P}(t)$ are then merged into an extended population $\mathcal{E}(t)$, from where the best P individuals are selected to become part of the new population $\mathcal{P}(t+1)$.

To accelerate algorithmic convergence, the initial population is generated based on a Stochastic Nearest Neighbor (SNN) heuristic [6]. The notion of *nearest neighbor* in the SNN heuristic is based not only on the distance to the last added node but also on the increment on the cost of recourse of the resulting tour. Phenotypic clones are eliminated and replaced by randomly generated solutions, leading to a diversified initial population with good quality solutions.

The crossover operator is based on the GVR crossover proposed by Pereira et al. [8], in which a child inherits all the traits (routes) from one parent and a small portion of the genetic material (subroute) from the other parent (donor). The subroute is randomly selected from the donor and inserted into the child at the lowest insertion cost. To speed up the procedure, the insertion cost is calculated taking into account only the deterministic part of the objective function (1). Duplicate nodes are eliminated from the child preserving those in the inserting subroute.

The mutation operator, known as *inversion mutation*, reverses the visit order of all nodes in a randomly selected subroute. Since the expected cost of recourse of a route is not symmetric in both directions [4], the inversion mutation diversifies the search not only in terms of route structure but also in terms of traveling direction.

3.2 Reparation and fitness evaluation

The genetic operators may generate infeasible solutions in terms of the distance constraint. To repair individuals, the GA uses a procedure based on the *split* mechanism proposed by Prins [9]. Split requires the GVR genotype to be transformed into a chromosome without route delimiters; that is, a single permutation of nodes (see top of Figure 1). From the chromosome, an auxiliary graph G' is built and used to find the optimal partition of the permutation into feasible routes. The directed graph $G' = (\mathcal{N}', \mathcal{A})$ is composed of the node set $\mathcal{N}' = \{0, i_1, \dots, i_k, \dots, i_n\}$ and the arc set \mathcal{A} . Nodes $i_1, \dots, i_n \in \mathcal{N}' \setminus \{0\}$ and node 0 is an auxiliary node. Arc $(i_k, i_{k+n_r}) \in \mathcal{A}$ represents a feasible route r with length l_r starting and ending at the depot and traversing the sequence of client nodes from i_{k+1} to i_{k+n_r} . The split procedure consists of finding the set of arcs (i.e., routes) along the shortest path connecting 0 and i_n in G' . Figure 1 illustrates the split procedure. A GVR individual containing an

infeasible route ($L = 60$) is transformed into a single chromosome, followed by a partition into two feasible routes using split.

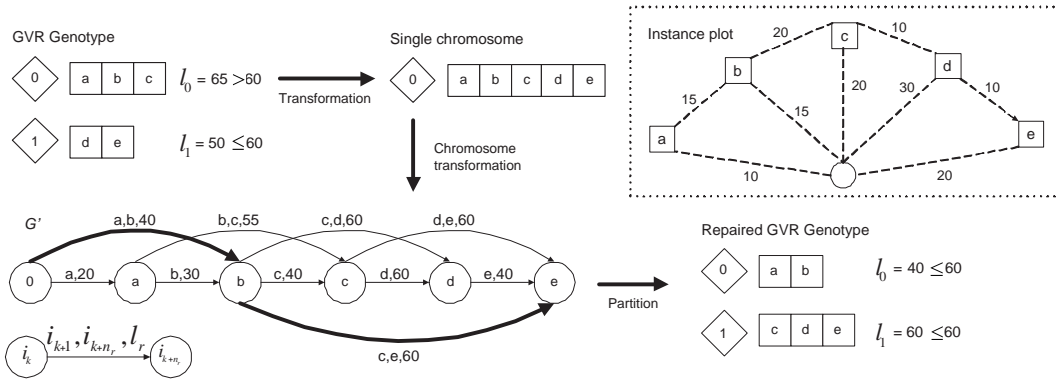


Figure 1: Split procedure for individual reparation

The fitness of an individual is the objective function (1). Under the proposed recourse action, the cost of recourse of a route is given by $E[G_r(x, \xi)] = \sum_{k=1}^{n_r} 2d_{(0, i_k)} \times p(z(i_k) = 1)$, where $p(z(i_k) = 1)$ is the probability of failing while visiting node i_k . On the multi-compartment scenario, estimating the exact value of $p(z(i_k) = 1)$ is a complex and time consuming process, for what the GA uses the take-all approximation (TAP) derived in [5]; that is

$$p(z(i_k) = 1) = \sum_{j=0}^{k-1} \left[\prod_{p=1}^m F(i_{j+1}, i_{k-1}, p) - \prod_{p=1}^m F(i_{j+1}, i_k, p) \right] \times p(z(i_j) = 1) \quad (2)$$

where $F(i_{j+1}, i_{k-1}, p)$ is the cumulative probability that the total demand for product p collected between nodes in position $j + 1$ and $k - 1$ in route r , does not exceed the capacity of the compartment Q_p .

To evaluate the fitness of an individual, two strategies are proposed. The first strategy (split) consists of applying the split procedure followed by the evaluation of the objective function of the repaired solution. On the other hand, the second strategy (s-split) consists of a stochastic variant of the split procedure that simultaneously repairs and evaluates an individual. Following the same principle of its deterministic counterpart, s-split partitions the single chromosome into feasible routes by finding the shortest path in the auxiliary graph G' . However, the weight of each arc $(i_k, i_{k+n_r}) \in \mathcal{A}$ is given by $l_r + E[G_r(\mathcal{R}, \xi)]$ instead of l_r . Figure 2 illustrates the s-split procedure on the same example shown in Figure 1.

4 Computational experiments

After fine tuning the set of parameters (P , T , p_c , and p_m) for each evaluation strategy, the GA was tested on a large set of randomly generated instances. Problems with 50, 100, and 150 client nodes and 3 products were generated with the following characteristics. For each

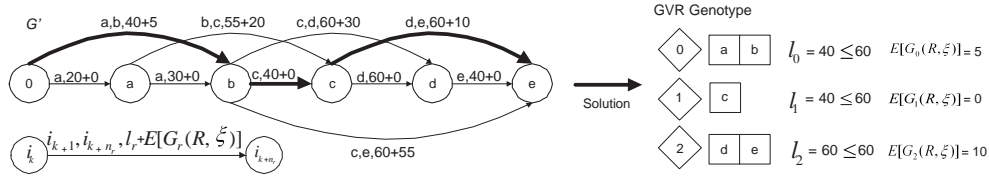


Figure 2: The s-split procedure for individual reparation and evaluation

demand, the value of $\mu_{i,p}$ was randomly generated within $[20,100]$; whereas the value of $\sigma_{i,p}$ was set by randomly selecting a coefficient of variation $\sigma_{i,p}/\mu_{i,p} \in \{0.1, 0.2, 0.3\}$. To set the capacity of the compartments, the tightness ratio $(\sum_{i \in \mathcal{N} \setminus \{0\}} \xi_{i,p})/Q_p$ was set at two different values, namely, 5 and 15. Thirty instances were generated for each combination of number of clients and tightness ratio. For each of the 180 ($3 \times 2 \times 30$) problems, the maximum distance L was randomly set as 3 or 4 times the distance from the farthest node to the depot.

To compare the evaluation strategies, four different metrics were calculated: first, the average (avg.) improvement with respect to the best solution in $\mathcal{P}(0)$ (after split or s-split); second, the average gap with respect to the best solution found by the GA; third the number of best solutions found; and fourth, the CPU time. Table 1 summarizes the results.

Instance type		Avg. improvement		Avg. gap to best		Best solutions		CPU time (s)*	
clients	tightness	split	s-split	split	s-split	split	s-split	split	s-split
50	5	14.17%	12.11%	1.75%	0.52%	10	20	89	322
50	15	12.45%	11.22%	0.57%	0.78%	19	11	98	310
100	5	19.45%	17.84%	1.43%	0.63%	12	18	223	1279
100	15	13.40%	12.39%	1.77%	0.54%	11	19	256	1266
150	5	20.11%	18.58%	1.25%	1.02%	12	18	438	3310
150	15	14.96%	12.32%	0.60%	0.74%	15	15	408	2329
Average/Total		15.76%	14.08%	1.23%	0.70%	79	101	-	-

* Intel Xeon Woodcrest 5120 processor with 4 Gbytes of RAM, running Windows Server 64 bits.

Table 1: Comparison of split vs s-split evaluation strategies for the MC-VRPSD

The results show that both evaluation strategies lead the GA to an average improvement of 14.92% over the best solution in the initial population. A closer look to Table 1 shows that this improvement is larger on instances with low tightness ratios. A plausible explanation, is that on such instances fewer failures are expected. Hence, there are more routing alternatives that avoid failures or it is possible to allocate them in convenient positions. The iterative process of the GA exploits this condition better than the single-pass SNN, generating significant improvements in the cost of recourse. On the other hand, the results show that the improvement is less significant on instances with low number of clients. As in other VRP variants [3] it is likely that in the MC-VRPSD the construction heuristics, like the SNN, are very competitive in small instances, letting to the GA small margins for amelioration. Focusing on the comparison between the evaluation strategies, a further analysis shows a slightly better performance of s-split over split. Out of the 180 instances, s-split reported the best solution in 101 vs 79 for split. Note, however, that in terms of solution accuracy this disparity might not be significant since the difference in the average gap with respect to the best solution is less than 1% (1.23%-0.70%). In contrast, in terms of CPU time the results clearly show that

split outperforms s-split. Hence, provided the tradeoff between accuracy and efficiency, split seems to be better for the overall performance of the GA.

5 Conclusions

This paper presents a GA for the MC-VRPSD that couples components proven to be effective in deterministic VRPs with two fitness evaluation strategies that account for the stochastic nature of the problem. Experiments conducted on a set of random instances show that the genetic effort is responsible for improvements of 14.92% (on average) with respect to the best solution in the initial population, found by a multi-start construction heuristic. The execution time varies from 1.48 to 38.81 minutes depending on the instance size and the selected evaluation strategy. Research currently underway includes the implementation of local search procedures and population management techniques specially tailored to enhance the proposed GA.

References

- [1] P. Avella, M. Boccia, and A. Sforza (2004): “Solving a fuel delivery problem by heuristic and exact approaches”. In: *European Journal of Operational Research* **152**, 170–179.
- [2] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto (2006): “Hybrid metaheuristics for the vehicle routing problem with stochastic demands”. In: *Journal of Mathematical Modelling and Algorithms* **5**, 91–110.
- [3] A. El Fallahi, C. Prins, and R. Wolfler Calvo (2008): “A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem”. In: *Computers and Operations Research* **35**, 1725–1741.
- [4] M. Gendreau, G. Laporte, and R. Séguin (1996): “Stochastic vehicle routing”. In: *European Journal of Operational Research* **88**, 3–12.
- [5] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco (2008): “Approximating the expected cost of recourse on a multi-compartment vehicle routing problem with stochastic demands”. Tech. Rep. 08/03/AUTO, École des Mines de Nantes, France.
- [6] J. E. Mendoza, C. Guéret, B. Castanier, A. L. Medaglia, and N. Velasco (2008): “Constructions heuristics for the multi-compartment vehicle routing problem with stochastic demands”. In *XIV Latin Ibero-American Congress on Operations Research (CLAIO)*.
- [7] J. Oppen and A. Løkketangen (2008): “The livestock collection problem”. In *21st European Conference on Operational Research*.
- [8] F. B. Pereira, J. Tavares, P. Machado, and E. Costa (2002): “GVR: a new genetic representation for the vehicle routing problem”. In: *Proceedings of AICS 2002 - 13th Irish Conference on Artificial Intelligence and Cognitive Science*.
- [9] C. Prins (2004): “A simple and effective evolutionary algorithm for the vehicle routing problem”. In: *Computers and Operations Research* **31**, 1985–2002.