Recourse Actions for Capacitated Vehicle Routing Problems with Stochastic Demand

Operations Research and Quantitative Logistics Master Thesis
Econometrics and Management Science

Erasmus University Rotterdam
Erasmus School of Economics

Isabel Nobrega
Student ID: 579698
Supervisor: Dr. Twan Dollevoet
Second Assessor: Dr. Olga Kuryatnikova
Date: August 8, 2021

Contents

1. Introduction

The vehicle routing problem has been studied extensively since first introduced in 1959 by Dantzig and Ramser. It considers the problem of determining a set of routes for a fleet of vehicles such that every customer is visited only once and costs are minimised. In addition to its basic formulation, several variations of the problem have also been studied. Examples of these variations include the capacitated vehicle routing problem where the vehicles all have a maximum amount of goods they can travel with, and the vehicle routing problem with time windows where each customer has to be served within its respective time window.

When solving vehicle routing problems, one or more features of the problem are assumed to be fixed. These could be the number of goods demanded, the travel times between locations or even the presence of customers at the time of delivery. In real-life problems this is of course not always true, and what was a deterministic situation becomes a stochastic vehicle routing problem. Indeed, we often see reports in the news about disruptions of the normal flows of transportation. Examples include containers lost at sea or the recent obstruction on the Suez canal. These are often distant accidents that fortunately have little or no impact on our lives as individuals. Very different of course was the covid-19 pandemic in which virtually all of us were profoundly affected.

In the context of vehicle routing problems with stochastic demand, it is interesting to consider for example Labad et al. (2021), where the shortage of toilet paper in supermarkets, one very puzzling, almost comical, effect of the pandemic is discussed. This gives just one example of how supply chains can be disrupted to the extent of causing almost social panic. This "stochastic demand" is of course a very exceptional situation, that would require equally exceptional and unpractical measures to overcome. But in general, being able to handle unexpected events in a delivery process is a very important capability if any of these very common situations occur:

- There is a road accident that forces the route to be replanned.
- The vehicle has a technical failure.
- The business accepting the goods had to close due to sickness.
- The customer decides to change the amount of the order at the last minute, for example, when the delivery happens.

In this thesis we will focus on the capacitated vehicle routing problem with stochastic demands. When demand is subject to uncertainty, routes must be defined (at least partially) before the real demand values of the customers become known. Companies might for example have access to some statistical information about the demands in order to define these routes. In some situations, a vehicle may not have sufficient stock available to meet a customer's demand upon arrival, in which case a route failure is said to have occurred. Recourse actions can be applied in order to amend these failures. The action to take when a failure takes place has strong implications on the satisfaction of customers, as well as on the profitability of suppliers. In this thesis the effect of choosing different recourse actions is studied.

Our numerical experiments show that it is possible to improve on the simplest recourse action (detour to depot) by introducing more sophisticated recourse actions. In particular, re-optimizing the routes when failures occur provides significantly better results compared to using routes that are completely planned *a priori*.

The remainder of this thesis is organised as follows. In Section 2 the problem is defined and some notation is introduced, followed by a discussion of the relevant literature in Section 3. Section 4 introduces our proposed recourse actions. Next, a description of the data sets is provided in Section 5. Finally, Section 6 presents the results obtained and Section 7 provides some concluding remarks.

2. Problem description

We are given n customers, v vehicles and their maximum capacities Q. Furthermore, we know the distance c_{ij} and the time t_{ij} (in seconds) it takes to travel from location i to location j. The demand of each customer i follows a uniform distribution $U(a_i, b_i)$ and has an expected demand value equal to d_i. Our objective is to find a set of routes such that the total travelling distance of the vehicles is minimized, while still satisfying the following constraints:

1. Each customer is visited exactly once
2. Each route starts and ends at the depot
3. Each route has a total demand less than or equal to Q
4. Each route has a duration less than or equal to T, which corresponds to the work shift of the drivers.

In addition, a few assumptions are made:

1. The demands of the customers are only revealed upon arrival at their location
2. The service time for each customer is negligible
3. All vehicles leave the depot at the same time
4. Communication between the depot and the vehicles is available at all times, which will be very important in order for us to be able to adapt the routes as information regarding the customers' demands becomes available

When applying recourse actions, the duration of the planned routes can change. An example of this is when a route failure occurs and a detour to depot is implemented. The duration of the route will then increase by the time required to travel to the depot and travel back to the customer where the failure happened. For that reason, a penalty is incurred when the duration of a route exceeds T. In real life situations this might be the case, since the employees of a company would probably have to receive overtime pay for extra hours of work.

3. Literature review

The vehicle routing problem is an NP-hard problem, and hence determining an optimal solution for medium and large sized instances within a reasonable amount of time might be an impossible task. Therefore, many authors turn to heuristics as opposed to exact algorithms (Toth and Vigo, 2002). The vehicle routing problem with stochastic demands has been the most studied version of the stochastic vehicle routing problems according to Oyola et al. (2018), followed by the vehicle routing problem with stochastic travel times, whereas the most commonly adopted recourse action has been the detour to depot policy. This may be due to the fact that this recourse action is very easy to understand and to implement.

In Juan et al. (2011) the capacitated vehicle routing problem with stochastic demands is solved as a capacitated vehicle routing problem. First, the stochastic demands are replaced by deterministic demands set to the expected values of each customer demand. In addition, part of the vehicle's capacity is reserved as safety stock. This provides a baseline solution whose quality can be evaluated by means of a Monte Carlo simulation. The expected total costs due to possible route failures are estimated by applying a detour to depot policy whenever a failure occurs. Similarly, the reliability of each route is estimated by calculating the probability that a vehicle fails to serve every customer on its route.

Gauvin et al. (2014) formulate the CVRPSD as a set partitioning problem and the recourse action chosen is a detour to depot policy whenever the demand is larger than the remaining capacity of the truck. In the particular case where the vehicle becomes exactly depleted when serving a customer, the vehicle goes to the depot to replenish and then returns to the same customer. This is done because the authors believe that the increased complexity of calculating the expected failure cost will probably not justify the savings of returning to the next customer in the route in case a vehicle gets exactly depleted (instead of returning to the customer that the driver failed to service). A Poisson distribution is used to model the customers' demand.

In Zhu et al. (2014) a paired cooperative reoptimization strategy is developed for a pair of vehicles. It is considered that the customer demands follow a uniform distribution. As recourse actions detour to depot and a partial reoptimization strategy are applied. This is done by assuming that the two vehicles can communicate and adapt their planned routes in real time, while important information such as the remaining capacities and customers left to be visited is known to both vehicles. The service times of the customers are assumed to be negligible.

Some authors do not take any actions at all when failures occur. In Chepuri and Homem-de

Mello (2005), if a failure occurs, the respective vehicle is dismissed and returns to the depot. The failed customer along with the possible remaining customers that were part of the route are simply not served, and in turn a penalty is paid for each unserved customer. Demands are assumed to follow a gamma distribution.

It is also possible to determine a set of routes that is feasible for all possible demand outcomes belonging to a bounded uncertainty set, and consequently no recourse actions need to be considered. In Sungur et al. (2008), the solution found optimizes the worst-case scenario since it is assumed that all customers can attain their highest possible demand simultaneously. The authors show that the solution protects against unmet demand while incurring a small additional cost in comparison to the deterministic optimal solution.

In general, these studies choose one particular recourse action (mostly the detour to depot recourse action because of its easy implementation) and aim at achieving the best possible initial solution that minimizes the total expected travel costs. In other words, the goal is to obtain a solution that minimizes the total cost of the initial routes, and at the same time also minimizes the expected costs of executing a particular recourse actions in the second stage of the problem. In order to achieve this, the initial solution will tend to be "pessimistic" in the sense that it has to be able to cope with the higher possible amount of demands placed by the customers.

This thesis takes a different approach because we will not try to come up with the best performing initial routes. The initial solution obtained will likely incur in more failures than if we were to take the stochastic demands into account before the initial routes are created. Instead we will mainly focus on studying and comparing the performance of different recourse actions.

4. Methodology

In order to study the effects of different recourse actions, a starting solution is required. Therefore, the problem is first solved in a deterministic way. This will provide the baseline solution that will be used to study the effect of stochastic demands. Then, in order to assess the different recourse actions, a Monte Carlo simulation generates random demand values for each customer. These real demands are expected to differ from the demands that were initially assumed as fixed, and therefore some routes will have a lower total demand and others will have a higher total demand. Some of which might even fail if their total demand exceeds the maximum capacity of the trucks.

We are of course interested in the cases that do fail. Four different recourse actions are applied separately, and individually, in an attempt to amend or prevent possible route failures. The cost function is also separately evaluated in order to give us insights into which recourse action is more appropriate in which situation.

4.1 Initial solution

We first solve a deterministic capacitated vehicle routing problem by assuming that the demands of the customers are given in advance and are equal to d_i. To come up with an initial set of routes, the following mixed integer programming formulation is used.

$$\max \quad \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} s_{ij} x_{ij} \tag{1}$$

$$\begin{aligned}
\text{s.t.} \qquad \sum_{j=1}^{n} x_{0j} &= v && && (2)\\
\sum_{i=0, i \neq j}^{n} x_{ij} &= 1 && \forall j \in \{1, ..., n\} && (3)\\
\sum_{j=0, i \neq j}^{n} x_{ij} &= 1 && \forall i \in \{1, ..., n\} && (4)\\
y_i + d_j x_{ij} - Q(1 - x_{ij}) &\leq y_j && \forall i, j \in \{1, ..., n\}, i \neq j && (5)\\
z_i + t_{ij} x_{ij} - T(1 - x_{ij}) &\leq z_j && \forall i, j \in \{1, ..., n\}, i \neq j && (6)\\
d_i \leq y_i &\leq Q && \forall i \in \{1, ..., n\} && (7)\\
t_{0i} \leq z_i &\leq T - t_{i0} && \forall i \in \{1, ..., n\} && (8)\\
x_{ij} &\in \mathbb{B} && \forall i, j \in \{0, ..., n\}, i \neq j && (9)
\end{aligned}$$

This formulation is based on the CVRP model introduced by Borčinová (2017), where instead of minimizing the costs c_{ij} of traversing arc (i, j), the savings s_{ij} are maximized. For every pair of customers i and j the savings s_{ij} for joining the cycles $0 \to i \to 0$ and $0 \to j \to 0$ are calculated, where $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. The binary variables x_{ij} indicate if arc (i, j) is traversed in the optimal solution. Constraint (2) ensures that exactly v trucks leave the depot. Constraints (3) and (4) make sure that all customers are visited exactly once. Auxiliary variable y_i represents the load of the truck after leaving customer i, and so constraint (5) guarantees that the total demand of every route does not exceed Q.

Additional constraints (6) and (8) have been added to ensure that the routes do not exceed the maximum duration T. The auxiliary variable z_j indicates how much time has gone by from the moment a route departs from the depot until its arrival at customer j.

The model proposed by Borčinová (2017) and referred to as the modified assignment formulation is a very fast formulation due to the way it eliminates sub-tours (that is, cycles that do not go through the depot). It is shown to take just a few seconds to solve instances with up to 23 customers. We therefore make use of the formulation presented above, and a time limit is given to find an initial solution. After this time limit the best solution found is retained. Even though the instances used in this thesis contain much more than 23 customers, the formulation is expected to still deliver a good solution within a certain time limit.

4.2 Recourse actions

This section introduces the four recourse actions mentioned above. In general, a recourse action (called a recourse policy by some authors) is a corrective measure applied in case of a route failure. Many recourse actions are possible, with different degrees of implementation complexity and performance. A recourse action is a fundamental feature of a delivery route subject to uncertainty. Without instructions on how to proceed in case a demand cannot be met, the vehicle driver would have no clear indication of what to do next. In order to optimise resources it is therefore required to have at least one recourse action prepared so that the behaviour of the delivery process can be properly managed.

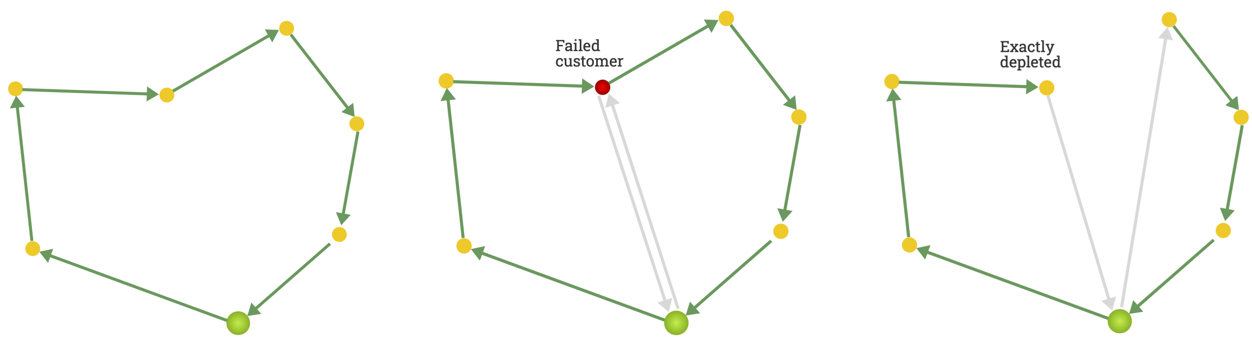
4.2.1 Detour to depot

The simplest recourse action that can be adopted is the detour to depot action, shown in Figure 1. There are two situations in which a detour to depot happens:

1. A truck arrives at a customer but the actual demand of the customer exceeds the truck's

current load. In this situation, in order to amend the failure, the truck driver leaves his current load with the customer and goes to the depot to restock. Afterwards, the truck driver returns to the customer it had failed to fully serve and delivers the remainder of the order to the customer. The truck then continues on to the next customer in the planned route. See Figure 1b).

2. The truck's load is exactly depleted after successfully serving a customer. It is clear that if this customer is the last customer of the route, the truck driver has finished his shift and simply returns to the depot. On the other hand, it can also happen that there are still customers in the route left to serve. In that case, we know with certainty that a failure will take place at the next customer and hence a preventive detour to depot is undertaken in order to avoid a pointless trip to the next customer. The truck driver travels to the depot to restock, and afterwards moves on to the following customer. See Figure 1c).



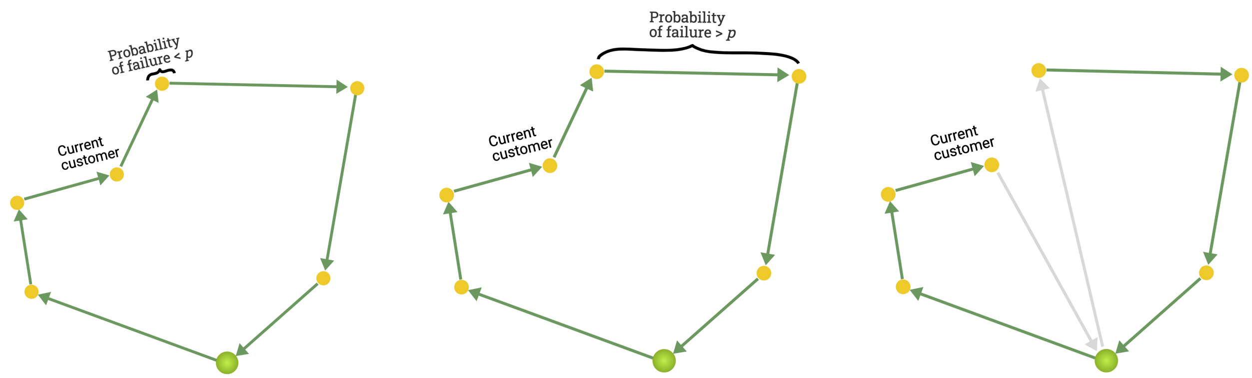
(a) Planned route **(b)** Failure **(c)** Exactly depleted

Figure 1: Detour to depot: a) the planned route as calculated by the deterministic formulation; b) the truck cannot (fully) serve the order at this customer; c) the truck is empty after serving this customer

This recourse action focuses on each route independently, and only implements a detour to depot in case one of the previous situations happens. This means that the order in which the customers are visited per route (which is given by the initial planned routes) is never changed. In terms of computation time, the impact of implementing this recourse action is negligible.

4.2.2 Preventive restock

Performing a preventive restock is another recourse action that can be used. This recourse action aims to prevent trucks from travelling to a customer that will most probably result in a failure. This is shown in Figure 2.



(a) Checking $k = 1$ **(b)** Checking $k = 2$ **(c)** Preventive restock

Figure 2: Preventive restock: a) current customer i was just served and there is a probablitity smaller than p of failing at customer $i + 1$; b) the probability of failing at customer $i + 2$ is larger than p, therefore c) the algorithm may choose to do a preventive restock at customer i

Each time a truck is finished serving a customer in its route, this recourse action calculates the probability of a failure occurring at the subsequent k customers in the route (see Appendix A.1 for details on how this probability is calculated). We start with $k = 1$, and increase k by 1 until a customer is found for which the probability of failure exceeds a certain value p, or until all remaining customers in the route have been covered.

If all the remaining customers in the route can be served without incurring a failure with a probability larger than p, then no preventive restock is done, and the truck proceeds to the next customer. In the case that they cannot all be served, a preventive restock is considered at the current location. The cost of performing a preventive restock at the current customer is compared with the cost of performing a detour to depot at each one of the next k customers. In the example depicted in Figure 2 we have that $k = 2$.

If the cost of restocking at the current customer is cheapest and the sum of the average demands of all the remaining customers left to serve in the route does not exceed $q \cdot Q$, then a preventive restock is chosen. The parameter q has been introduced in order to make sure that a preventive restock is not implemented at the beginning of a route, when a truck still has a lot of customers left to serve. This way we can reduce the risk of performing multiple trips to the depot in the same route.

Figure 2c) shows a situation where it is cheaper to perform a preventive restock after customer i than to wait for a failure to occur and having to implement a detour to depot at either one of the next $k = 2$ customers. In this example, the sum of the average demands of the 3 remaining customers in the route is smaller than $q \cdot Q$ and so a preventive restock is executed.

There is, of course, always a risk. Since we can never be certain of a failure (unless p is set to 1), it could happen that a failure would actually occur at the last customer of the route from

Figure 2. In that case we can see that the last customer of the route is closer to the depot than our current customer, and therefore it would have been cheaper to wait for the failure to occur. There might even be no failure at all. Therefore the parameter p has to be chosen carefully. If it is set too low, there will be more unnecessary preventive restocks and costs might be higher. If it is set too high, this recourse action will only implement a preventive restock when it is almost certain that a failure will occur. Similarly, we can also analyse qualitatively the effect of changing parameter q. Higher values of q will increase the number of preventive restocks performed because they will also be allowed at earlier stages of the route.

Just like the detour to depot recourse action, the initial planned routes are not changed when applying the preventive restock recourse action. Furthermore, this recourse action also has an extremely low running time since calculating the probabilities can be done almost instantly. Therefore, the amount of time that a truck driver has to wait to know if it will go to the depot to replenish or not is negligible.

4.2.3 Reoptimization

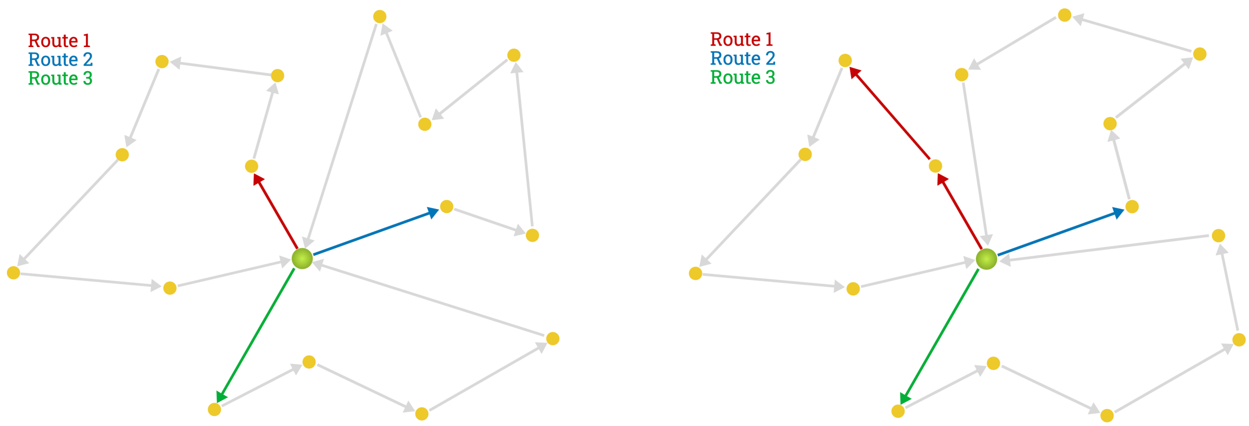
Contrary to the previously discussed recourse actions, the reoptimization recourse action takes into account not only the current truck and the remaining customers in its route, but customers from other routes as well. In general, a customer can be in one of the following classes:

1. Already served customers
2. Not yet served, but a vehicle is already traversing the arc to that customer (referred to as "fixed" customers)
3. Not yet served, and no truck en route

So, whenever a truck successfully serves a customer, the customer to which the truck heads to next becomes "fixed". A fixed customer, although not having been visited yet, is excluded from the reoptimization procedure, as well as all the customers that have already been served.

Upon arrival at a customer, two options exist: either a failure happens or not. In case the truck is able to meet the current customer's true demand, all the customers that have not been visited yet (excluding fixed customers) are reoptimized over again. When optimizing over these customers, the true demand of the current customer is used instead of its average demand. Therefore, it might happen that the customer to which the current truck was planning to go to next differs from the customer where he will actually go to according to the routes obtained from the reoptimization method. This could be the case, for example, if the true demand of the current customer is much larger than its average demand. It might be a better option to let another truck serve one of the customers in the current truck's route since it now has a much larger total expected demand, or even to change the route entirely.

In case the demand of a customer cannot be met, no reoptimization is done. The truck is dismissed and the driver returns to the depot. The customer that was not able to be served along with the possible remaining customers meant to be served by the current truck will be reassigned to other trucks still operating. These customers will be included in the next reoptimization round. An exception to this is when the current truck is the last operating vehicle, in which case a detour to depot is performed by the current truck in order to serve the current customer and the remaining unvisited customers.



(a) Initial routes

(b) Reoptimized routes

Figure 3: Reoptimizaion

An example of the reoptimization procedure is given in Figure 3. In Figure 3a), the initial routes obtained for a particular instance are shown. The trucks all leave the depot at the same time, and when they do, the initial arcs become fixed. Fixed arcs are shown in Figure 3 as being colored. Since we are given the travel time between each pair of customers, we know the exact time at which each truck driver will arrive at its respective fixed customer. As can be deduced from Figure 3a), the first truck to arrive at a customer is the red one. Upon arrival, the true demand of that customer becomes known and the truck driver is able to meet that demand. We proceed to perform a reoptimization, now using the known demand of the customer instead of the previously used average demand. Because the initial conditions changed, and the problem space is now smaller, the solver might be able to find a better solution compared to the initial one. Figure 3b) shows the solution obtained after reoptimizing the routes. The red truck heads to its next customer, which is now different from the one he would go to if the previously planned routes were followed, and the arc becomes fixed so that it can not be changed anymore.

The reoptimization procedure is done by using the formulation presented in Section 5.1. Since the search is performed in real time, while a truck driver is waiting to know which customer to serve next, a time limit of 30 seconds is given to the solver to find a solution.

4.2.4 Limited reoptimization

The limited reoptimization recourse action differs from the reoptimization recourse action in the sense that it does not perform a reoportimization every time a vehicle is finished serving a customer. Instead, this recourse action only reoptimizes the routes in the following two situations:

1. A failure happens. In this situation the truck leaves its current stock with the customer and goes to the depot. To determine what happens next, the reoptimization method considers both options of the truck returning or not to service. The truck can be dismissed, in which case the reoptimization procedure returns a configuration where the unserved customer and the possible remaining customers that were supposed to be served by the current truck are assigned to some other truck(s). Likewise, if returning this truck to service delivers a better solution, the truck goes to the depot to restock and the reoptimization method computes a solution where all the remaining customers left to serve (excluding fixed customers) are shared by all the available trucks (so, including the current truck). The routes are optimised without regard to the old state, so that there is no memory of what the previous planning was.

2. A truck is able to successfully serve a customer but is likely to fail on the next one. If the remaining load of the current truck is sufficient to satisfy the average demand of the next customer in its route, the truck simply continues on to the next planned customer. If it is not enough, the truck driver goes to the depot and a reoptimization is done according to the algorithm described in the previous item.

Just like in the reoptimization recourse action, we use the formulation introduced in Section 5.1, with an additional constraint that allows for an extra vehicle to be added to the solution or not. Again, a time limit of 30 seconds is given to the solver to find a solution. The difference between the two recourse actions is that in the limited one we only reoptimize the routes whenever a failure happens or is likely to happen at the next customer. These situations will most probably happen near the end of each route, which means that the first time a reoptimization is performed, a lot of customer have already been served. The less customers a capacitated vehicle routing problem has, the faster it can be solved, meaning that 30 seconds will likely be more than enough time for the solver to find a good solution.

4.3 Monte Carlo simulation

After having obtained the initial routes, each recourse action is implemented separately in order to improve the planned routes. The vehicles start by following the initial planned routes when departing from the depot, and depending on which recourse action is implemented, these routes can be altered or not. We make use of a Monte Carlo simulation that will generate random demand vectors by randomly picking for each customer i a demand value contained in its respective interval (a_i, b_i). Afterwards, for a fixed number of iterations, the costs of applying each corrective policy are compared.

In the detour to depot and the preventive restock policy each route is adapted independently, meaning that we never assign a customer from one route to another. Thus there is no need need to keep track of which vehicle arrives first at which customer during our simulation. We can simply monitor one vehicle at a time, while keeping track of its capacity, until all vehicles have been checked.

For the limited reoptimization and the reoptimization recourse action we do need to keep track of time. In these recourse actions, the current solution is altered as a whole, and so not only the current vehicle is affected, but all the other routes are adapted as well. For these policies, the simulation program moves to the vehicle that is the next one to arrive at a customer, and continues until all customers have been served. This can be done because we know all times t_{ij}, and so each time routes are optimized, we can determine the times at which the vehicles arrive at their respective fixed customers.

5. Data

The CVRP data sets generated by De Smet (2014) are used to test the methods presented in this thesis. The following characteristics make these datasets particularly useful:

- The goal of De Smet was to create realistic datasets and therefore these instances consist of real locations for cities, towns and subtowns of Belgium. The distances between every pair of locations are given as the real road distances as opposed to the euclidean distances most commonly used in datasets available on the internet.

- De Smet studied the difference between solving the capacitated vehicle routing problem using the real road distances and using euclidean distances. When euclidean distances are used to solve the problem, certain routes are selected; interestingly, these are different from the routes obtained when using real distances. It turns out that the "euclidean solution" is suboptimal compared to the "real solution" in the sense that the total distance covered by the trucks is smaller in this case than the distance obtained using the euclidean metric.

- The instances contain the time it takes to travel between every pair of customers in seconds.

Reasonable vehicle capacities and customer demands were also added. In order to perform the simulation, the given deterministic demand values $d_i \in \mathbb{N}$ of each customer are changed to stochastic demands D_i. Each customer demand follows a discrete uniform distribution $U(a_i, b_i)$ where:

$$E[D_i] = d_i \tag{10}$$

$$a_i = \lfloor d_i \cdot (1 - \epsilon) \rfloor \tag{11}$$

$$b_i = \lceil d_i \cdot (1 + \epsilon) \rceil \tag{12}$$

In Appendix A.2 we show that (10) is always satisfied regardless of the value that is chosen for ϵ. For the simulation an ϵ of 0.4 will be used, unless stated otherwise.

Finally, regarding the size of the problem instances, the two smallest instances generated contain 49 and 99 customers, while the remaining three instances contain 499 or more customers, with the biggest one having 2749 customers. Only the first two instances will be used for testing so that the problem can be solved within reasonable time.

6. Results

In this section, the methodology is tested on the two smallest instances generated by De Smet (2014). For both instances, the initial solution obtained when running the formulation proposed in Section 5.1 for a certain time limit is compared to the optimal solution of the deterministic version of the problem. That is, the optimal solution when the demands of each customer i are assumed to be fixed and equal to d_i, and no maximum duration on the routes is imposed. We have used a method by Nóbrega (2021) to obtain the optimal solution of both instances. The time limit given to the solver to find an initial solution differs according to the size of the problem instance. For the smaller instance we allow 30 seconds, and for the larger instance 1 minute.

After that, the results obtained when implementing each one of the proposed recourse actions are reported and discussed. Compared to the detour to depot and the preventive restock recourse actions, the reoptimization recourse action has a very long simulation time because we are reoptimizing the routes multiple times, potentially as many times as there are customers. In the limited reoptimization recourse action, the reoptimization procedure is performed much less often, and for that reason, the simulation time is reduced significantly compared to the simulation time of the reoptimization recourse action. We therefore first compare all four recourse actions by running the Monte Carlo simulation 50 times. Afterwards, 1000 iterations are performed, testing only the detour to depot, preventive restock and limited reoptimization policies. The simulations are performed using the same initial routes and demand realizations.

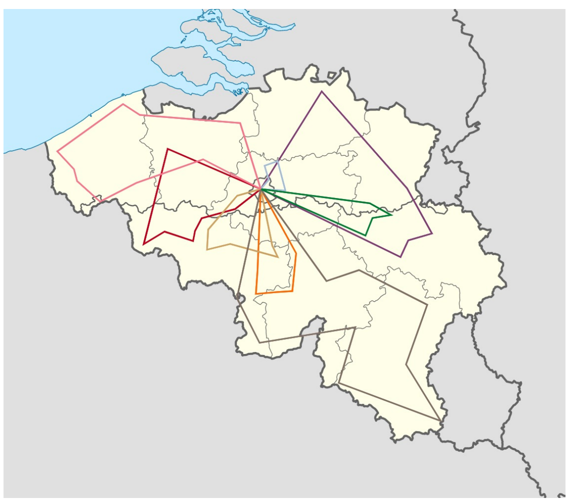
The work shift T of the drivers is set to 8 hours. If it happens that a driver ends up working for longer than T, a penalty of 1 is incurred for every minute extra exceeding T. Parameters p and q are chosen as follows: for the small instance, the probability p is set to 70%, and q is set to 70%, leading to a preventive restock only being possible if a failure is at least 70% likely to occur at some next customer in the route, and if the cumulative demand of all the remaining customers in the route does not exceed 70% of the maximum capacity Q of a truck. As for the big instance, p is set to 50% and q is set to 60%. The reason as to why these values were chosen is explained in Section 6.4 where the effect of choosing different values for p and q is analysed. Finally, the effect of changing the parameter ϵ is also studied in Section 6.5.

6.1 Instance 1

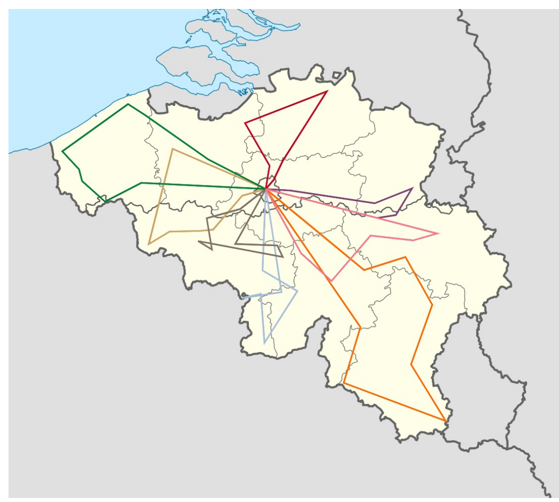
The first and smallest instance tested contains 49 customers and a depot, and there are 8 vehicles available. As mentioned above, the time limit to find an initial solution using the formulation presented in Section 5.1 is set to 30 seconds.

6.1.1 Initial solution

The optimal solution of this instance has a total cost of 2046.7, and the respective routes are shown in Figure 4a). Curiously, these optimal routes all have a duration shorter than 8 hours, even though this was not required by the method used to find the optimal solution. This makes the comparison more meaningful. The initial solution obtained which will be used to test the recourse actions has an objective value of 2184.8, which is a value less than 7% higher than the optimal solution. The resulting routes can be seen in Figure 4b), and are quite different from the optimal routes. Overall the formulation performs well on this instance, given that it achieves a good initial solution within 30 seconds. In Appendix A3 a list of the customers visited by each truck is given for both the optimal and initial routes.



(a) Optimal routes



(b) Initial routes

Figure 4: Solution for instance 1

6.1.2 Recourse actions

As mentioned above, we start with comparing the recourse actions by performing 50 iterations of the simulation. Table 6.1 shows the average costs of applying each recourse action when using the initial routes depicted in Figure 4b). From the table we can see that the detour to depot recourse action is, on average, the most expensive recourse action, followed by the preventive restock recourse action. The reoptimization recourse action performs, on average, slightly better than the preventive restock recourse action, whereas the limited reoptimization recourse action performs best. In 46 out of 50 iterations, at least one route failure occurred as can be seen in Table 6.2. At least one preventive restock was executed in 5 iterations, and an unnecessary preventive restock was never performed. This means that, if in any of these 5 iterations a preventive restock would not have been chosen, then there would eventually have been a route failure. When taking only these 5 iterations into account, the average cost of the preventive restock recourse action is 2432.4, while the average cost of the detour to depot recourse action is 2454.2. Even though there is a noticeable difference for these 5 iterations, in the other 45 iterations the two recourse actions behaved in the same way, and so their total costs were identical. That is why their average costs, shown in Table 6.1, are very similar.

The simulation times were also included in Table 6.1. As mentioned earlier, the reoptimization recourse action takes a very long time, namely almost 8 hours just to finish 50 iterations.

Table 6.1: Average costs of the recourse actions for the small instance, 50 iterations

Recourse action	Average cost	Total simulation time
Detour to depot	2425.5	6 milliseconds
Preventive restock	2423.4	47 milliseconds
Reoptimization	2418.1	7.8 hours
Limited reoptimization	2377.4	174 seconds

Table 6.2: Number of failures for the small instance, 50 iterations

	Iterations
Failures	46
Preventive restock	5
Unnecessary preventive restock	0

Table 6.3 provides more accurate values of the average cost of the detour to depot, preventive restock and limited reoptimization recourse actions, calculated over 1000 demand simulations. The total simulation times are also included. As mentioned above, due to its long simulation time, the reoptimization recourse action is not included this time. Once again, the limited reoptimization recourse action performs best. As can be seen in Table 6.4, at least one route failure occurred in 959 iterations. The preventive restock recourse action chose to execute at least one preventive restock in 79 iterations. The average cost of the final routes of these 79 iterations is 2448.0 when using a preventive restock recourse action compared to the average cost of 2482.8 when using a detour to depot recourse action. Even though, on average, it was indeed cheaper to implement the preventive restock recourse action in these 79 iterations, a replenishment is still not executed very often given that 1000 iterations were performed. Therefore the difference in the average cost of the detour to depot and the preventive restock recourse actions is small, as can be concluded from Table 6.3. Additionally, an unnecessary preventive restock was never executed, meaning a preventive restock

was never executed in a solution where a route failure would actually not occur.

Table 6.3: Average costs of the recourse actions for the small instance, 1000 iterations

Recourse actions	Average cost	Total simulation time
Detour to depot	2408.3	50 milliseconds
Preventive restock	2405.5	259 milliseconds
Limited reoptimization	2352.3	62 minutes

Table 6.4: Number of failures for the small instance, 1000 iterations

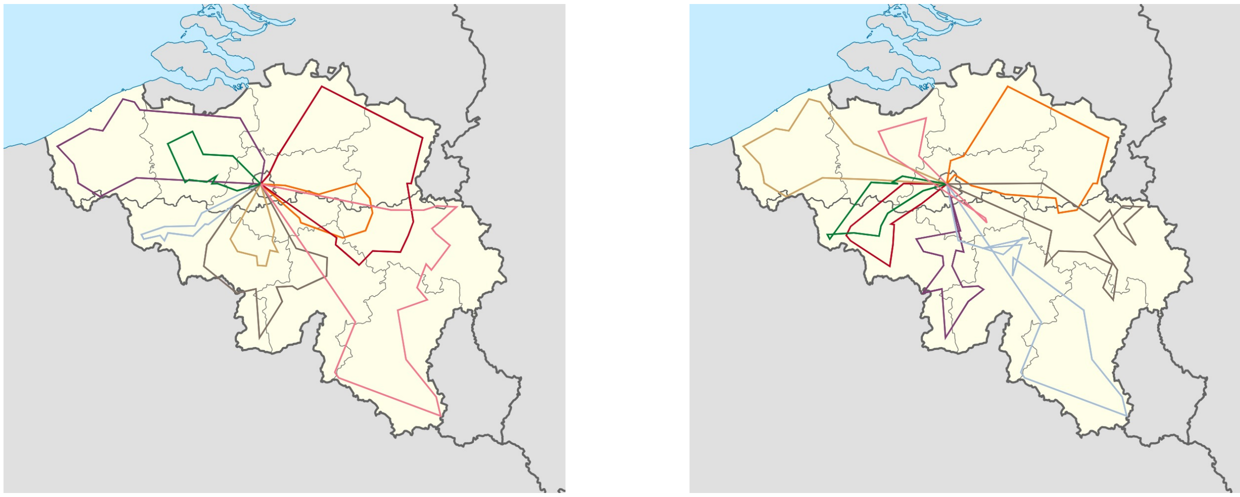
	Iterations
Failures	959
Preventive restock	79
Unnecessary preventive restock	0

6.2 Instance 2

The second and largest instance tested contains 99 customers and a depot, and there are 8 vehicles available. This instance is a superset of the previous instance, meaning that the granularity of the locations has increased. In order to find an initial solution, and given the increase in the number of customers compared to the first instance, the time limit for the solver is set to 60 seconds.

6.2.1 Initial solution

The optimal solution for the second instance is shown in Figure 5a), and has a total cost of 2320.634. Again, all routes part of the optimal solution take less than 8 hours to complete. The formulation used to obtain the initial routes does not perform as well on the second instance, since these routes are not as close to optimal compared to the initial routes of the first instance. The initial solution is presented in Figure 5b), and has a cost of 2644.1, a result that is around 14% above the optimal value. Appendix A4 contains the routes shown in Figure 5.



(a) Optimal routes (b) Initial routes

Figure 5: Solution for instance 2

6.2.2 Recourse actions

We now compare the different recourse actions, by performing 50 iterations of the Monte Carlo simulation. Table 6.5 shows the average costs of applying each recourse action separately. Similar to the results of the small instance, a much better result is achieved when the limited reoptimization recourse action is adopted. Again, the reoptimization recourse action performs slightly better than the preventive restock recourse action. However, the difference between the detour to depot recourse action and the preventive restock recourse action is now more noticeable. As can be seen in Table 6.6, at least one route failure occurred in 43 iterations, and a preventive restock was executed in 13 iterations. The average cost of the preventive restock recourse action in these 13 iterations is 2940.5, while the average cost of the detour to depot recourse action is 3071.6.

Table 6.5: Average costs of the recourse actions for the large instance, 50 iterations

Recourse actions	Average cost	Total simulation time
Detour to depot	2889.2	13 milliseconds
Preventive restock	2855.2	322 milliseconds
Reoptimization	2850.4	26.4 hours
Limited reoptimization	2815.2	7 minutes

Table 6.6: Number of failures for the large instance, 50 iterations

	Iterations
Failures	43
Preventive restock	13
Unnecessary preventive restock	0

Now, 1000 demand simulations are executed comparing only the detour to depot, preventive restock and limited reoptimization recourse actions. Table 6.7 shows the average costs of these recourse actions.

Table 6.7: Average costs of the recourse actions for the large instance, 1000 iterations

Recourse actions	Average cost	Total simulation time
Detour to depot	2877.6	99 milliseconds
Preventive restock	2858.0	3 seconds
Limited reoptimization	2812.6	152 minutes

Table 6.8: Number of failures for the large instance, 1000 iterations

	Iterations
Failures	908
Preventive restock	342
Unnecessary preventive restock	39

While in the previously tested instance a preventive restock was executed at least once in 79 iterations, it is now executed at least once in 342 iterations, as is shown in Table 6.8. On average, the final routes obtained in these 342 iterations have an objective value of 2969.5 when applying the detour to depot recourse action, and a total cost of 2907.1 when using the preventive restock recourse action. In 39 iterations a preventive restock was executed when in reality it was not needed. In other words, if no preventive restock would have been executed in any of these 39 iterations, meaning that the the truck drivers would simply have continued their planned routes without going to the depot to replenish, there would not have been a route failure anyway. So, in reality there was no need to restock.

6.3 Performance of the recourse actions

The preventive restock recourse action has, on average, a better performance than the detour to depot recourse action. However, for the small instance, a preventive restock is not executed very often, and so the difference in average cost between the detour to depot recourse action and the preventive restock recourse action is almost insignificant. As for the larger instance, the difference between the average cost of these two recourse actions is more noticeable.

On the other hand, the reoptimization recourse action performs worse than the limited reoptimization recourse action. The difference between these two approaches is that in the limited case the routes are modified each time a failure occurs, whereas in the reoptimization situation they are modified at each delivery, so, much more often. These solutions are likely to be less robust than the ones from the limited reoptimization because they will tend to follow very closely the average demands from which they are calculated, and adjust poorly to the effective, stochastic demands. This would be similar to the mechanism we know from machine learning, where a solution that follows the training data too closely will tend to generalise poorly for test data. In this analogy, training data corresponds to the average demands, whereas test data corresponds to the stochastic demands.

It is interesting to mention that when using the reoptimization or limited reoptimization recourse action, it so happens that in some iterations the final routes generated by these recourse actions have an even lower cost than the cost of the initial routes (in spite of the stochastic nature of the demands). This mostly occurred in iterations where the initial planned routes would not incur in a failure if they had not been changed.

6.4 The effect of parameters p and q

In order to understand the effect of p and q, simulations were performed on a grid of (p, q) values, where $p = 0.4, 0.5, \ldots, 0.9$ and $q = 0.7, 0.8, \ldots, 1.0$. The cost functions of the preventive restock recourse action were calculated for each pair of values p and q (using the same 1000 simulated demand values for each pair) and the result is shown in Figure 6. Each function represents a different value of q. Due to the way these parameters influence the outcome, we would expect some kind of minimum in the cost function within these regions. This minimum will depend on the particular instance, meaning that if the the initial routes change, or if the locations or the average demands of the customers change, the logistics company should calculate new optimal values of p and q and use these new values in the preventive restock algorithm.

For instance 1, we can conclude by looking at the top graph of Figure 6 that the probability that delivers the best results is $p = 0.7$. It turns out that, no matter what value is chosen for q, using this probability always delivers the same result, namely, a result of 2402.9. We have therefore chosen to set both values to 70% in our simulations of instance 1.

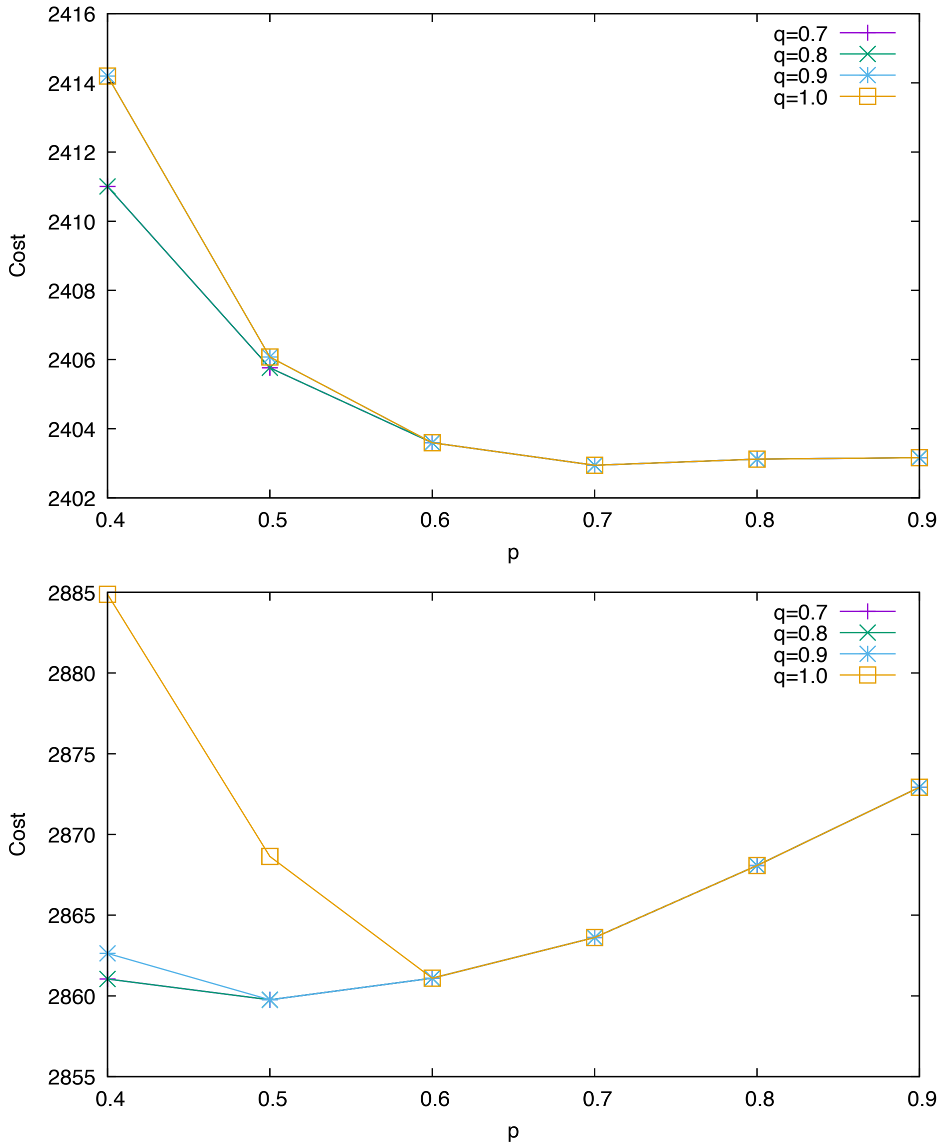


Figure 6: Comparing results for different values of p and q. The image above represents the results for instance 1, and the image below the results for instance 2.

From the bottom graph of Figure 6 it is clear that for instance 2 the best probability to choose is $p = 0.5$. The total cost of the preventive restock recourse action when using $p = 0.5$ and any value of q between 60% and 90% is 2859.7. The total cost remains the same for all those values of q, except for $q = 1$. In this case, the total cost is higher and equal to 2868.6. This is because of route number 6 from the initial routes of instance 2 (see Appendix. A4). After serving the first customer of this route (customer 75), the probabilities of a failure happening at the next customers are calculated. Eventually, a customer is found for which the probability of failing is larger than or equal to $p = 50\%$, and a preventive restock is considered because customer 75 is located very close to the depot. Afterwards, it is checked if the sum of the average demands of the remaining customers in the route, which is equal to 229, is larger than $q \cdot Q = 1 \cdot 250$. This is not the case and can actually never happen because we never plan routes that have a sum of (average) demands larger than Q. So, a preventive restock is chosen. However, a failure is now very likely to happen later in the route since 229 is a very high average demand. In the other cases for $p = 50\%$, for example when $q = 90\%$, we have $q \cdot Q = 0.9 \cdot 250 = 225$. Since in this case, (and the other ones as well) 229 is a larger sum, the preventive restocking is not performed. Therefore a higher average cost is associated to $p = 0.5$ and $q = 1$. For the simulations of instance 2, we use $p = 50\%$ and $q = 60\%$.

6.5 The effect of parameter ϵ

The simulations in this study were executed with a value of $\epsilon = 0.4$. This means that the stochastic demands were allowed to vary 40% around the average demands. However, this was just a sample value. In reality, the width of the uniform distributions could be higher, and that is why this parameter ϵ was introduced in the model. If we increase ϵ, demands will have a wider distribution, and we should expect that the benefits of using the more sophisticated recourse actions become more noticeable. This is indeed what happens, as is shown in Figure 7. Here, for each instance, parameter ϵ varies from 0.2 to 1.0, and 200 demand simulations have been performed for each value of ϵ. The initial routes shown in Figure 4b) and 5b) have been used for testing.

For each instance, three cost functions of applying each recourse actions are shown (left y axis), respectively detour to depot (DTD), preventive restock (PR) and limited reoptimization (LR) in addition to two ratios (right y axis). The ratios show the gains in cost function for PR and LR against DTD.

We can see from the graphs of Figure 7 that for both instances, as ϵ increases, the average cost of applying DTD, LR or PR also increases. This makes sense because the more uncertainty present, the more the expected demands will differ from the true demands, and therefore more failures will occur which translates into higher costs. Furthermore, LR shows consistently higher gains compared to PR. We note that the higher the value of ϵ, the more important it is to apply the limited reoptimization recourse action, compared to the other two recourse actions. Whereas gains are around 1-2% for $\epsilon = 0.2$, they grow to about 4% for $\epsilon = 1.0$, which can translate into substantial operational gains for the company.

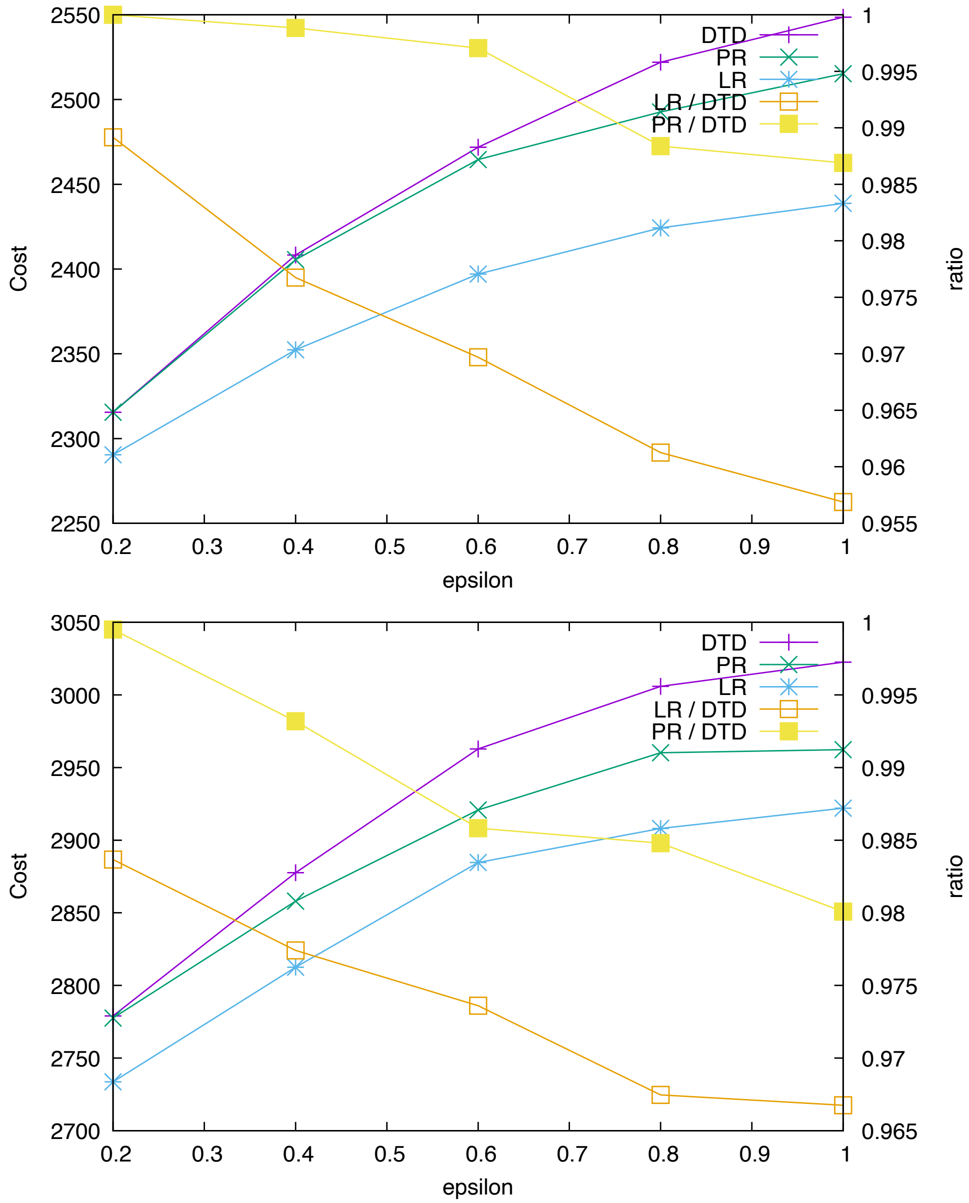


Figure 7: Comparing results for different values of ϵ. The image above represents the results for instance 1, and the image below the results for instance 2.

7. Conclusion

In this thesis we considered the capacitated vehicle routing problem with uncertainty in customer demand. The uncertainty was modeled using a uniform distribution, with parameterised width, and a formulation was adapted from the literature to assist in the calculation of delivery routes. Four recourse action were developed in order to prevent or amend route failures, ranging from the simplest (detour to depot) to the most complex (reoptimization). Monte Carlo simulations were then performed, supported by a combination of a Java program and a commercial solver (CPLEX), with the aim of comparing the performance of the different recourse actions.

The results show that the limited reoptimization recourse action can achieve substantial improvement compared to the detour to depot recourse action. In particular, the wider the demand distribution is (meaning, the higher the uncertainty), the more relevant it is to apply the limited reoptimization recourse action, with improvements up to 4% in the cost function.

7.1 Suggestions for further study

Even though the preventivive restock, the limited reoptimization and the reoptimization recourse actions already show improvements compared to detour to depot recourse action, there are particular areas where we believe that further exploration could be worthwhile.

7.1.1 Time constraints

Currently, the routes in the limited reoptimization and reoptimization recourse action are not allowed to have a duration exceeding T. This could be a potential limitation of these recourse actions. Incorporating a more sophisticated way of modeling the time costs could be a better option. It might happen that a cheaper solution is available if the routes are allowed to exceed T. Even though one or more truck drivers would have to extend their shift for a bit longer, the cost of these routes (including the penalty costs of exceeding T) could be cheaper than the cost of the routes when dismissing these truck drivers so that their shifts take less than T.

7.1.2 Impact of the initial routes

In this thesis the same initial routes are used for all simulations. It would be interesting to know just how much impact the initial routes have on the average costs of a company. If more time was spent in this study finding an initial solution for the vehicles, would the average cost of implementing each recourse action also have delivered noticeable better results? Or would preparing the initial solution for uncertainty in demand be a better option? In most approaches adopted in the literature, the initial routes are calculated in such a way that they end up being more robust to uncertainty, meaning that failures are less likely to occur in these routes. These initial routes will have higher costs compared to initial routes that do not take any uncertainty into account. But do these additional costs compensate in the end?

7.1.3 Optimal location for the depot

The performance of the recourse actions depends on the location of the depot compared to the locations of the customers. In our instance which is located in Belgium, the depot is located in Brussels and is therefore in the central portion of the country. The detour to depot recourse action here performs relatively well because the trucks are always within a short distance from the depot. For countries where this is not the case, like for example in Austria, where the capital is located close to the northeastern border of the country, it is expected that the detour to depot recourse action (assuming a depot in Vienna) would perform worse. Using a simple model where countries are circles of radius R, having the depot at the center would give an average distance between customers and the depot of $2R/3$ compared to $32R/(9\pi)$ (see Stone, 1991) for a depot at the border, almost twice the distance, and therefore almost twice the additional cost in case of failure.

Having this in mind, a company may therefore consider relocating the depot in order to reduce costs. The methods and algorithms we developed in this thesis could be easily extended or modified to tackle this problem: we would simply have to estimate the cost function (and the recourse actions' performance) for each proposed location and determine the best solution.

7.1.4 Trucks sharing stock

An interesting feature of an additional recourse action would be the ability for the trucks to share their stock in real time.

Given the random nature of the actual demands, it could happen that one truck has lack of goods, while another has excess goods, and their locations at a certain point in time are such, that it would be possible for them to meet and balance their stocks. In effect, one of the trucks would behave like a "movable depot" and share its excess stock with the other one. This recourse action, while more complex to develop and implement, has the potential to deliver additional savings.

8. References

Borčinová, Z. (2017). Two models of the capacitated vehicle routing problem. Croatian Operational Research Review, 8; 463–469.

Bradley, D., Gupta, R. (2002). On the Distribution of the Sum of n Non-Identically Distributed Uniform Random Variables. Annals of the Institute of Statistical Mathematics, 54, 689–700

Chepuri K., Homem-de Mello T. (2005). Solving the vehicle routing problem with stochastic demands using the cross-entropy method. Ann. Operations Research 134(1):153–181

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. Management Science, 6(1):80–91.

De Smet (2014): http://www.optaplanner.org/blog/2014/09/02/VehicleRoutingWithRealRoad Distances.html

Juan A., Faulin J., Grasman S., Riera D., Marull J., and Mendez C. (2011). Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. Transportation Research 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).

Labad J., González-Rodríguez A., Cobo J., Puntí J., Farré JM. (2021). A systematic review and realist synthesis on toilet paper hoarding: COVID or not COVID, that is the question. PeerJ.

Nobrega A. (2021). Transportation and Scheduling Assignment 1: Vehicle Routing (Unpublished). Erasmus University Rotterdam.

Oyola, J., Arntzen, H., and Woodruff, D. L. (2018). The stochastic vehicle routing problem, a literature review, part I: models. EURO Journal on Transportation and Logistics, 7(3):193–221.

Stone, R. (1991). Technical Note—Some Average Distance Results. Transportation Science 25(1):83-90

Sungur, I., Ordóñez, F., and Dessouky, M. (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. IIE Transactions, 40(5):509–523.

Toth, P. and Vigo, D. (2002). The vehicle routing problem. SIAM.

Zhu L., Rousseau L-M., Rei W., Li B. (2014). Paired cooperative reoptimization strategy for the vehicle routing problem with stochastic demands. Computers and Operations Research, 50:1–13

A. Appendix

A.1 About the distribution of the sum of n non-identical discrete uniform random variables

From Bradley et al. (2002) we know that the distribution can be calculated either by computational means or by closed form. It is interesting to compare both approaches as this gives us insights into what is actually required to evaluate this probability.

From Theorem 2 in the paper we know that the probability mass function of the sum of n uniform variables X_i in the set $\{-m_i, 1-m_i, \ldots, m_i-1, m_i\}$ is

$$g_n(p) = \frac{M}{2^{n-1}} \sum_{k=0}^{(n-1)/2} \frac{(-1)^k b_{2k}^{(n)}}{(n-2k-1)!} + \sum_{\vec{\epsilon} \in \{-1,1\}^n} \Big(2p + \sum_{j=1}^{n} (2m_j+1)\epsilon_j\Big)_+^{n-2k-1} \prod_{j=1}^{n} \epsilon_j$$

where

$$M := \prod_{j=1}^{n} (2m_j+1)^{-1}$$

and

$$b_{2k}^{(n)} = (-1)^k \binom{n+2k}{n} \sum_{m=0}^{2k} \frac{n}{n+m} \binom{2k}{m} \frac{1}{2^m(2k+m)!} \sum_{r=0}^{m} (-1)^r \binom{m}{r} (2r-m)^{2k+m}$$

The authors mention that the probability can be calculated directly using

$$g_n(p) := (\chi_1 * \chi_2 * \cdots * \chi_n)(p) = \sum_{k_1+k_2+\cdots+k_n=p} \prod_{j=1}^{n} \chi_j(k_j)$$

but that this formula is “inconvenient to apply” so we compared both approaches: their closed-form expression was evaluated with Mathematica, and a numerical estimate was calculated with Java.

While giving the same results, the Mathematica script takes much longer to compute the probabilities for large values of n. For example for $n = 12$ Mathematica takes around 10 seconds to compute the probabilities whereas the Java function finishes in around 70 milliseconds. For this reason the Java implementation was used in the simulations.

A.2 For integer demands, any real ϵ produces a symmetrical interval

We will show that (10) is always satisfied regardless of the value that is chosen for ϵ. In order for (10) to be satisfied, the following equation has to hold:

$$d_i - \lfloor d_i \cdot (1 - \epsilon) \rfloor = \lceil d_i \cdot (1 + \epsilon) \rceil - d_i \tag{13}$$

where the left term represents the width of the lower half of the uniform distribution interval and the right term represents the upper half.

Known properties of the floor and ceiling functions are, for integer m and real x

$$\lfloor m + x \rfloor = m + \lfloor x \rfloor \text{ and } \lceil m + x \rceil = m + \lceil x \rceil \tag{14}$$

In order to simplify (13), we note that $d_i \cdot (1 \pm \epsilon) = d_i \pm d_i \cdot \epsilon$. This can be written as $d_i \pm (m + x)$, for some $m \in \mathbb{N}$ and $x \in [0, 1[$. The left term in (13) is

$$\begin{aligned} d_i - \lfloor d_i \cdot (1 - \epsilon) \rfloor &= d_i - \lfloor d_i - d_i \cdot \epsilon \rfloor \\ &= d_i - \lfloor d_i - (m + x) \rfloor \\ &= d_i - \lfloor d_i - m - x \rfloor \\ &= d_i - [(d_i - m) + \lfloor -x \rfloor] \\ &= m - \lfloor -x \rfloor \end{aligned}$$

And the right term is

$$\begin{aligned} \lceil d_i \cdot (1 + \epsilon) \rceil - d_i &= \lceil d_i + d_i \cdot \epsilon \rceil - d_i \\ &= \lceil d_i + (m + x) \rceil - d_i \\ &= d_i + m + \lceil x \rceil - d_i \\ &= m + \lceil x \rceil \end{aligned}$$

For $x = 0$ we have

$$m - \lfloor -x \rfloor = m$$
$$m + \lceil x \rceil = m$$

and for $x \in]0, 1[$ we have

$$\begin{aligned} m - \lfloor -x \rfloor &= m - (-1) \\ &= m + 1 \\ m + \lceil x \rceil &= m + 1 \end{aligned}$$

In both cases we arrive at an equality. This completes the proof.

A.3 Routes of Instance 1

Optimal routes:

Route 1: 0, 1, 43, 49, 37, 28, 0

Route 2: 0, 13, 10, 41, 33, 38, 0

Route 3: 0, 17, 20, 19, 46, 0

Route 4: 0, 23, 16, 27, 8, 26, 32, 34, 0

Route 5: 0, 24, 18, 30, 21, 15, 40, 0

Route 6: 0, 29, 12, 42, 44, 39, 35, 2, 6, 4, 0

Route 7: 0, 47, 5, 45, 14, 0

Route 8: 0, 48, 3, 22, 11, 36, 7, 25, 9, 31, 0

Initial routes (obtained with a time limit of 30 seconds):

Route 1: 0, 5, 45, 37, 14, 46, 0

Route 2: 0, 12, 1, 42, 44, 39, 35, 2, 0

Route 3: 0, 13, 33, 6, 10, 4, 41, 0

Route 4: 0, 20, 31, 28, 19, 17, 0

Route 5: 0, 23, 16, 27, 26, 15, 40, 0

Route 6: 0, 30, 24, 18, 8, 21, 32, 34, 0

Route 7: 0, 48, 9, 25, 7, 36, 11, 22, 3, 0

Route 8: 0, 49, 43, 47, 29, 38, 0

A.4 Routes of Instance 2

Optimal routes:

Route 1: 0, 6, 44, 91, 22, 72, 14, 99, 21, 50, 65, 18, 62, 40, 0

Route 2: 0, 17, 93, 9, 94, 10, 28, 75, 55, 92, 0

Route 3: 0, 30, 29, 16, 69, 13, 89, 32, 97, 54, 52, 64, 0

Route 4: 0, 34, 38, 56, 61, 35, 74, 49, 86, 2, 57, 24, 27, 63, 0

Route 5: 0, 60, 39, 87, 26, 53, 85, 36, 48, 5, 0

Route 6: 0, 68, 42, 45, 8, 20, 12, 41, 82, 66, 83, 58, 23, 7, 76, 0

Route 7: 0, 80, 67, 3, 43, 37, 73, 79, 46, 19, 96, 47, 0

Route 8: 0, 90, 25, 81, 59, 15, 98, 95, 77, 71, 51, 31, 84, 33, 88, 11, 78, 70, 1, 4, 0

Initial routes (obtained with a time limit of 60 seconds):

Route 1: 0, 36, 85, 53, 82, 66, 83, 41, 12, 20, 8, 26, 87, 39, 60, 5, 0

Route 2: 0, 40, 38, 56, 61, 35, 74, 90, 10, 28, 55, 92, 34, 0

Route 3: 0, 48, 7, 63, 9, 76, 27, 58, 23, 33, 88, 11, 78, 70, 1, 4, 0

Route 4: 0, 68, 29, 30, 42, 45, 13, 89, 54, 67, 0

Route 5: 0, 73, 6, 44, 91, 22, 72, 14, 99, 21, 50, 65, 18, 46, 0

Route 6: 0, 75, 49, 25, 98, 95, 77, 15, 59, 81, 86, 71, 84, 31, 51, 2, 57, 24, 94, 0

Route 7: 0, 80, 64, 52, 16, 69, 97, 32, 79, 3, 37, 43, 0

Route 8: 0, 93, 17, 96, 19, 62, 47, 0