Ca' Foscari
University
of Venice

Master's Degree

in Data Analytics for Business and Society

Final Thesis

Reinforcement Learning for a Routing Optimization Problem.

Solving a VRP with a FedEx data set.

Supervisor
Ch. Prof. Raffaele Pesenti

Assistant supervisor
Ch. Prof. Denis M. Becker (NTNU)

Graduand
Angelica Ricci
Matriculation Number 878539

Academic Year
2023 / 2024

To my family, my friends and
Leonardo. You are the
nymph of my life.

Preface.

This thesis is the result of a project carried out during my academic experience with other two fellow students (Francesco Nardin and Beatrice Fabris) for the course Managerial Decision Making and Modelling (EM1407). My idea was to improve the results obtained using Reinforcement Learning and to critically compare the two solutions.

For this research, Artificial Intelligence (Chat GPT) has also been used as a helpful basis for the developed code in the second chapter. Moreover, it gave insights for our discussion and proposed new ways of conducting the research. The implementation of this tool is declared throughout the course of this thesis.

Abstract

This thesis wants to solve a Vehicle Routing Problem through the use of a non-traditional method, Reinforcement Learning. This is complemented by the resolution of the same problem through heuristic techniques and a deep analysis of the two implementations.

Firstly, the problem is solved with open-source tools provided by Google, mathematical and optimization functions. Subsequently, the same problem is solved through the development of an environment and the utilization of specific Reinforcement Learning algorithms. These generate the paths of the vehicles from the warehouse to the several customers by training an agent, which decides the actions to be taken. Lastly, an economic analysis of the two proposals is carried out concentrating especially on the new method.

The research shows that the traditional method optimizes the vehicles' routes but can work well only with small sets of non-real world data, as it faces several limitations. On the other hand, Reinforcement Learning models are more complex and can work with big sets of real world data. It must be said that this study needs further refinement to provide optimal solutions, as the ones offered are not the best ones. As a matter of fact, when trying to generalize unseen data, the model is not efficient enough. However, Reinforcement Learning remains a promising way of optimizing internal business processes, which requires additional resources and study to successfully complete its tasks.

Table of contents.

Introduction.

In the progressive panorama of optimization problems, the need for more accurate and productive solutions has led to a change over the implemented techniques. As a matter of fact, there has been a shift from the traditional greedy procedure to a more complex approach.

The aim of this thesis is to apply Reinforcement Learning (RL) to efficiently solve an optimization challenge, with an expert eye on a Vehicle Routing Problem (VRP) developed over a FedEx dataset. Nowadays delivery systems have become sophisticated since they provide ground-breaking technologies able to track the shipment whenever needed, they offer a 24/7 customer service and a higher level of the parcel security among other things. For this reason, companies operating in this sector require refined solutions to improve the decision-making process.

RL can be explained as a branch of Machine Learning (ML) that bases its application on rewarding intended performance (as described in [1]). It is mainly used to train ML models because – through an agent – it can face the several complications of the context in which it was created. This approach sees its application especially in the following fields: gaming, robotics, customized recommendations, management of the resources, trading and finance, healthcare and Natural Language Processing (NLP).

Under a business perspective, the logistic sector is intended to manage the life cycle of resources along the supply chain (as it is outlined in [2]). Today, e-commerce is the primary source of income for countless enterprises and this is because they innovated their supply chain logistics, making it easier for customers to buy and receive their purchases directly at home. Technologies including Real-Time Tracking (RTT), advanced route optimization and predictive analysis pushed digital transformation to make a huge transition in this industry. These mechanics will be further discussed in this thesis.

In the logistic and transportation framework, VRP can be seen as an integral component of resource management. Its primary purpose is to allocate the available resources to satisfy the clients and deliver the best possible service taking into consideration multiple constraints (such as minimization of the time and the traveled distance, available number of vehicles, position of the pick-up points with respect to the warehouse, etcetera).

In Chapter 1, the heuristic procedure used to solve the optimization problem with the FedEx dataset is explained and the Python code showed; of course, an explanation of how RL is going to be employed with the view to improve the result is given. Chapter 2 is the heart of this thesis, as it provides a resolution proposal of the challenge employing RL and Chapter 3 examines what discovered in Chapter 2 through a SWOT (Strengths, Weaknesses, Opportunities, Threats) and other types of economic analysis, evaluating the costs it takes and the benefits it can bring. The programming language of these projects is Python.

Through the study around RL, this research tries to offer concrete resolutions that may help in the improvement of the efficiency and effectiveness of real-world processes. In addition, the final analysis is intended to compare the advantages and disadvantages of RL with respect to the rule-based method already implemented.

Chapter 1. Towards Reinforcement Learning.

1.1 Introduction.

The company examined is FedEx Corporation, an American company founded in 1971 that works in the logistics and delivery sector, with a particular attention towards e-commerce and business services. As disclosed in [3], FedEx operates both in the domestic and international scenario and it furnishes a wide range of services among others, customer service and technical support. Its business is divided into different operational units, which are: FedEx Express (mainly known for its "*next day air service*" [4] in the US area especially), FedEx Ground, FedEx Freight, FedEx Logistics, FedEx Services, FedEx Dataworks and FedEx Office. It has almost 2,000 locations, around 530,000 employees, serves more than 200 countries and is listed on the stock exchange.

1.2 Description of the project.

For the project, it has been decided to only consider the circumscription of the city of Dallas (Texas) in which the FedEx trucks had to collect all the shipping items from the several pick-up points or shipping facilities. This, with the intention of delivering the parcel to the warehouse of the selected area.

As already stated in the Introduction, this type of problem is called VRP and its major goal is minimizing costs by optimizing the road traveled by all the existing trucks. This will lead to a decision-making process, in which the DMs (Decision Makers) must determine which truck will serve a specific set of clients (pick-up points).

[5] outlines the elements of a VRP:

- Depot, or also called warehouse. It is the "*home base*" of all the different locations.
- Customers, or shipping points. Each one has explicit needs that will be fulfilled by the DMs' decisions and more specifically, by the FedEx trucks.
- Vehicles, liable for the delivery (or the pick-up) to the clients (or from the clients).
- Costs and constraints.

A renowned variation of VRP is the VRPTW (Vehicle Routing Problem with Time Windows), which takes under consideration the customers' needs within a pre-established time window. A time window can be described as a time constraint in which the demand shall be served and, in this case, when each truck should leave the warehouse and when they should come back. This important element can be divided into two categories: soft and hard TW, where the only difference is permission to violate the constraint given some penalty costs [6]. Referring to this specific project, the DMs are setting a hard time window that every truck shall comply with.

The problem described can be disentangled either through exact algorithms or heuristic approaches able to offer approximate solutions. The latter technique is going to be studied and used in the first Chapter of this thesis.

Heuristics can be divided into three main categories which are constructive improvement and two-phase methods. Hereinafter the explanations and some examples [7]:

- Constructive heuristics are simple to implement and can rapidly yield to a solution, even if it may not be the optimal one. Several algorithms fall in this class (and each one of them can be either sequential or parallel), such as Nearest Neighbor, Insert, Saving and Sweep methods.
- Improvement heuristics exploits local search to find the local optimum, which they iteratively enhance. The disadvantage of this pattern is the risk of getting stuck in a local optimal point

that can't be adjusted to the overall solution. Two-opt, GENI and CROSS are three famous examples.

- Metaheuristics have a strong global search capacity that helps them in reaching the improvement. Inter-route and intra-route are the two main classes of this category.

The heuristic proposal rotates around CFRS (Cluster First, Route Second). It's a two-phase method that can decompose a sophisticated problem in two stages: cluster first and route second precisely. In the first phase, all the shipping points are clustered into various groups (clusters) according to the distance between each other; consequently, an optimized solution for each cluster is given.

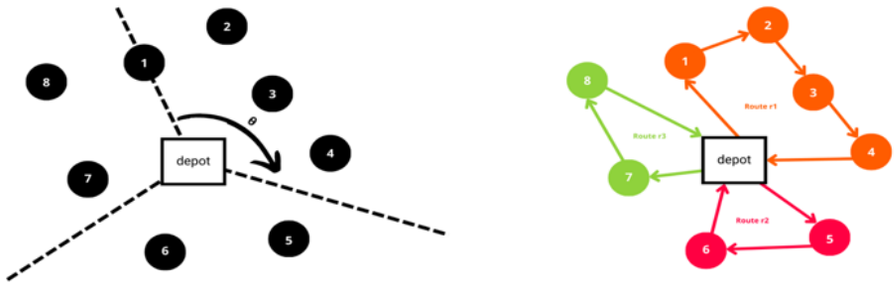


Fig. 1: F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, M. Yuan, 1st March, 2023. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. [Online]. Available: https://arxiv.org/abs/2303.04147v1 (recreated image).

The outcome highly depends on the position of the depot which sometimes can be decentered and leading to a weak performance. As suggested by [7], multiple "*reference points*" should be used instead of basing the solution only on the location of the single warehouse.

Another aspect that is worth mentioning is that VRP falls in the NP-hard problems (Nondeterministic Polynomial-time) class. Starting from the beginning, NP problems are decision problems that are possible to be tackled in polynomial time (and their solution is verifiable in polynomial time). On the other hand, a problem is NP-hard when "*all the problems in NP are polynomial time reducible to it*" [8]. The "*polynomial*" concept describes an algorithm with a running time – on inputs of size n – of $O(n^k)$ having $k > 0$. Questions solvable in polynomial time, are said to be "*trackable*": the real challenge is indeed that not all problems are possible to solve in polynomial time. For example, the Turing Halting Problem can't be solved by any algorithm created either by a human or by a computer.

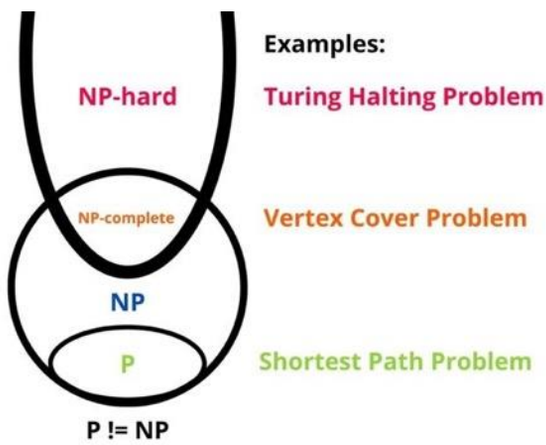


Fig. 2: V. Vaghela, *Medium*, November 23rd, 2020. Introduction to NP Completeness. [Online]. Available: https://vips3201v.medium.com/introduction-to-np-completeness-d3dfa771d994 (recreated image).

1.2.1 Problem statement.

As already anticipated in the previous paragraphs, this project's goal is to optimize the FedEx trucks route in the circumscription of the city of Dallas in the US. The path that they need to follow is generally used to pick up the shipping items from all the customers and bring them to the depot. To reach this objective, the average shortest path that minimizes the distance needs to be found.

One of the most important constraints of this problem is the time window in which the trucks can serve the customers. As a matter of fact, each vehicle must leave the warehouse at the beginning of the shift (02:30 pm) and go back to the central building at the end of the shift (06:30 pm); during this time (4 hours), all the shipping points must be assisted.

Here are the relevant components.

- Collected data until 2017 [9] [10] [11]:
    - In the US there are 48,821 FedEx facilities, of which 485 of them are in Dallas, 21 are warehouses and the remaining ones are shipping points.
    - Overall, FedEx possesses around 87,000 trucks in the US.
    - The trucks used are electric and their autonomy is about 400 km. They need a 2-hour charge to be completely loaded and they can carry a maximum of 1,000 kg.
    - The Dallas population consists of 1,341,103 inhabitants.
- Assumptions:
    - The total number of trucks is proportionally distributed among the number of facilities in the US, implying 864 trucks in the City of Dallas. These are divided in three categories, which are long-distance (30), picking-up service (278) and delivery service (556).
    - It's assumed all the trucks are electric (Brightdrop EV600 model [12]) since the company is planning to create an electric fleet by 2040.
    - The number of trucks for each warehouse is measured in proportion to the number of shipping points.
    - One third of the total population ships a package every day (around 447,755 per day) in equal measure among all the facilities (no competition is assumed).
    - One third of the daily packages are picked up by the FedEx trucks at the facilities. Two thirds are instead picked up at home.
    - The overall daily packages collected by FedEx trucks are 149,252 (around 400 packages per facility).
    - Each package's weight follows a normal distribution of $\mu = 1$ kg and $\sigma = 0.4$ (heavier packages may be picked up at home).
    - To load the shipment, each truck needs to stop for 40 minutes at every facility.
- Elements:
    - The agents are the FedEx logistic department (only decision maker) and the potential clients that can decide where and how many packages to deliver (it's assumed they opt for the closest facility).
    - Facilities, trucks, packages and warehouses are the entities of this project.
    - The distance is not computed at crow flies since trucks shall follow the roads (even if the first step of the resolution involves the implementation of the crow fly distance).
    - Traffic may slow down the travel time and sometimes, trucks' technical issues may occur.

[10] declares that FedEx will purchase only electric trucks after 2025 as it is committing to create a complete electric fleet by 2040. In fact, the project assumes that all the trucks involved in the delivery are electric (Brightdrop EV600 model). Drivers are responsible to charge their vehicles before the

beginning of the shift (02:30 pm), knowing that they will charge in 2 hours (from 0% to 100%). The top speed is limited to 65 mph (105kmh) and the speed limits of the streets; the mean cost of charging the trucks is around $0.44 and $0.55 per kWh. A fully charged carrier is around 50kWh and it would be able to run 400 km (the total amount should be $27.65) [12].

1.2.2 Mathematical model.

The parameters are:

- K = set of trucks (278).
- F = set of facilities (464).
- d_{ij} = travel distance from facility i to j.
- q_i = amount to pick up from facility i.
- s_i = service time at the facility i (40 minutes).

Decision variables, objective function and constraints are defined in [13] and here they are specified for this project. The decision variables are:

- $x_{ijk} \in \{0,1\}$ - 1 if truck k travels from facility i to facility j, 0 otherwise.

The objective function aim is to minimize the route that each truck needs to travel:

$$min(X) \sum_{i \in F} \sum_{j \in F} \sum_{k \in K} d_{i,j} x_{ijk}$$

The constraints, instead, are the following:

- Every truck must depart from the warehouse:

$$\sum_{j \in F} x_{0jk} \leq 1 \; \forall k \in K$$

- Every truck must come back to the warehouse:

$$\sum_{j \in F} x_{i0k} = x_{0jk} \; \forall k \in K$$

- All shipping points must be served by only one truck:

$$\sum_{k \in K} \sum_{j \in f, j \neq i} x_{ijk} = 1 \; \forall i \in F$$

$$\sum_{k \in K} \sum_{i \in f, i \neq j} x_{ijk} = 1 \; \forall j \in F$$

- The weight capacity of the truck is 1,000 kilos:

$$\sum_{i \in F} \sum_{j \in F, j \neq i} q_i x_{ijk} \leq 1{,}000kg \; \forall k \in K$$

- Every truck has 400 km of electric autonomy:

$$\sum_{i \in F} \sum_{j \in F} d_{ij} x_{ijk} \leq 400\ km\ \forall k \in K$$

- Every truck leaves the set of facilities F at least once (SEC, Subsequent Elimination Constraint) [14]:

$$\sum_{i \in F} \sum_{j \in F} x_{ij} \geq 1\ F \in \{1, 2, n\}, 1 \leq |F| \geq N - 1$$

The last constraint avoids the creation of the so-called "*subtours*". These can be represented in the following way:

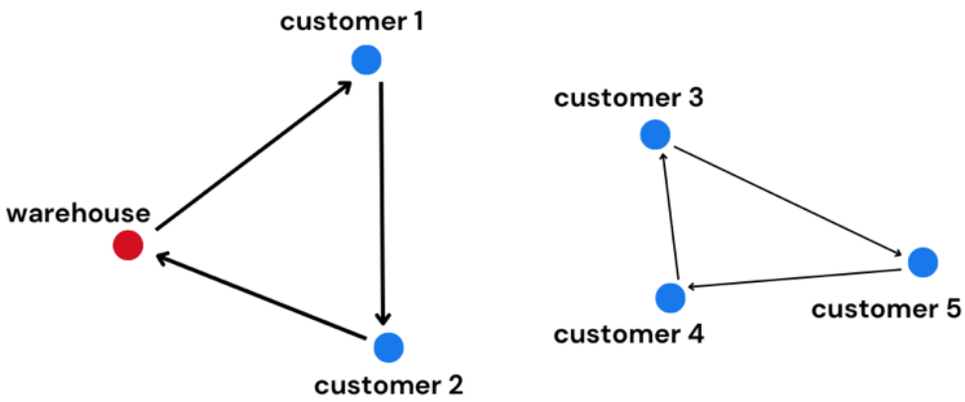


Fig. 3: subtours.

1.3 Development of the solution.

First and foremost, all the needed libraries are imported. They are essential for the good resolution of the problem, as some of them include data structures and numerical support. In addition, thanks to them it is possible to calculate distances, create interactive maps, create clusters, retrieve API Keys, access the solution to the problem and more.

After this important passage, data is imported as well and some modifications are made. Ineffective columns are dropped (e.g., '`X'`, '`Y`', '`Address2`' and '`Placement`') and the categories of all the facilities are set as '`warehouse`' and '`shipping_point`'. Moreover, all facilities' latitudes and longitudes are collected in a new data frame that will also contain the column '`type`' (with Boolean values, '`warehouse`' or '`shipping_point`') and all the shipping points are partitioned according to the distance from the closest warehouse.

The following task is creating a function (or better, two functions, one that returns a tuple and one that returns a single value) that computes the crow fly distance (in km) between the points in a geographical space. The following flowchart breaks down the '`geo_distance1'` function that is returning a single value.

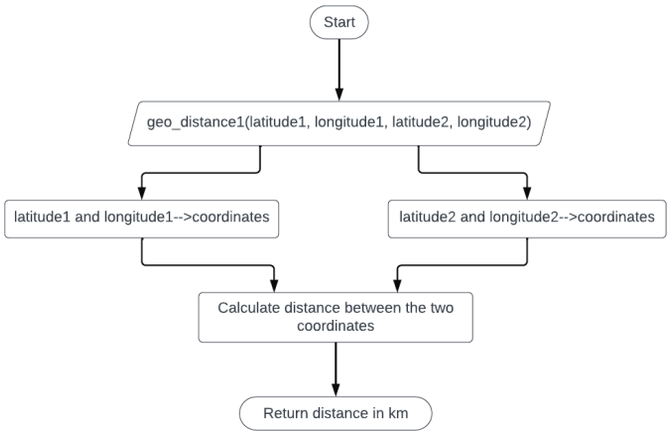


Fig. 4: geo_distance1() flowchart.

With the following steps, the shipping points are successfully assigned to the closest warehouse.

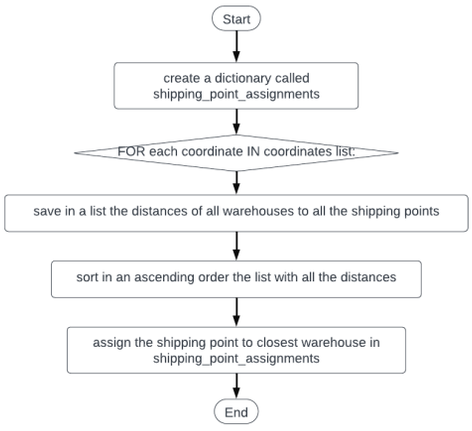


Fig. 5: for loop shipping_point_assignments flowchart.

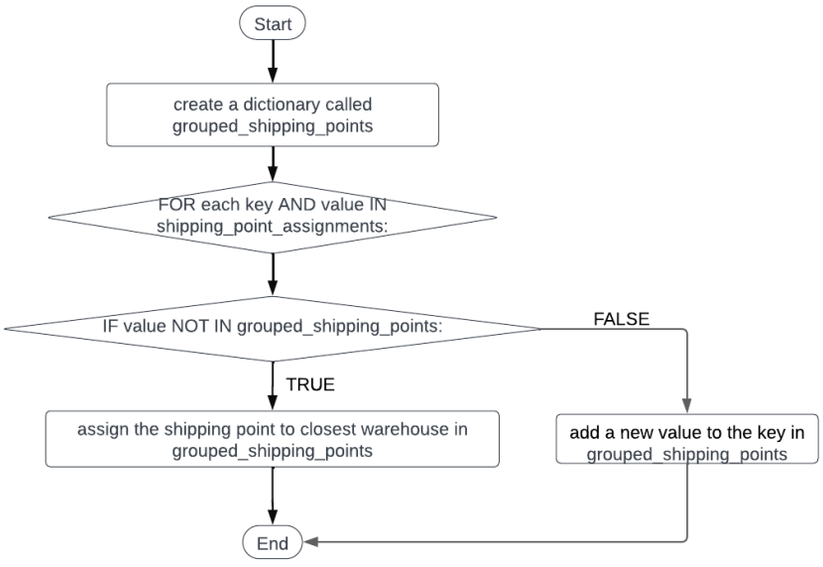


Fig. 6: for loop grouped_shipping_points flowchart.

The keys of '`grouped_shipping_points`" are stored in a list called '`w`' (warehouses), while their corresponding values in another list called '`sp`' (shipping points).

Group 1 is then extracted and the corresponding warehouse is appointed; this is done because it has been decided to work only with a specific set of shipping points (‘`group1`’) as previously remarked. To do so, ‘`group1`’ data frame is created taking into account the information regarding only the first group (latitude and longitude more specifically). Afterwards, a new row with the coordinates of the corresponding warehouse and a new column with the type of the location (shipping point), are added.

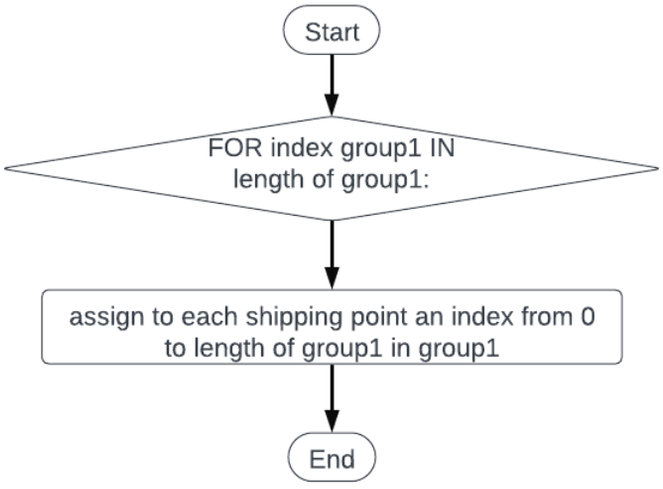


Fig. 7: for loop group1 flowchart.

Now it is possible to assign to the last row its appropriate type (warehouse).

Here Group 1 is displayed:

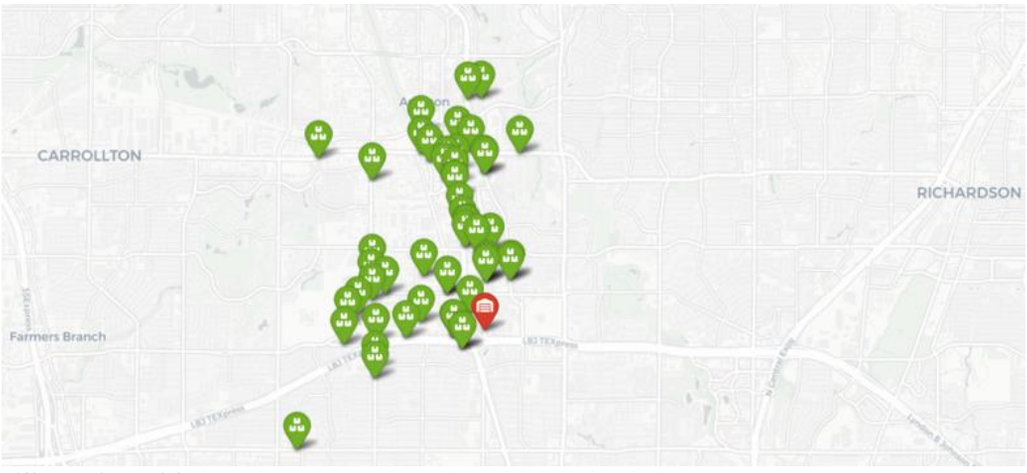


Fig. 8: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

The subsequent phase is to split the shipping points of Group 1 in 10 smaller clusters, in conformity with the distance from the warehouse: this will allow to solve the VRP. An important remark is that each sub-group is served by exactly three trucks.

The process is the following: ‘`group1_1`’ is a new data frame that contains only the shipping points of ‘`group1`’; ‘`coordinates_group1_1`’ contains the latitude and longitude of each shipping point; ‘`distances`’ calculates the distances of all shipping points with the ‘`pdist`’ function and the ‘`geodistance`’ metrics; ‘`distance_matrix`’ is the square distance matrix of ‘`distances`’.

A KMeans object is afterwards created and fitted to the matrix of distances with the purpose of clustering the first group in 10 sub-groups. Likewise, labels from 0 to 2 are assigned to each cluster. The below flowchart explains the process implemented to give to each location a specific label from 1 to 10 (10 sub-groups).

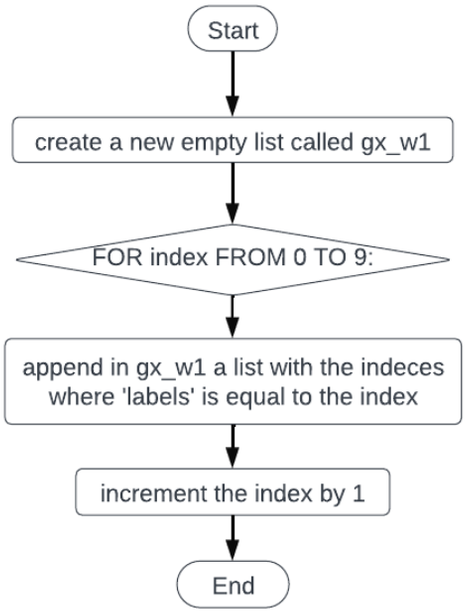


Fig. 9: for loop gx_w1 flowchart.

The following function adds a row with the information regarding the given warehouse in a new data frame which will be concatenated with another data frame (group).

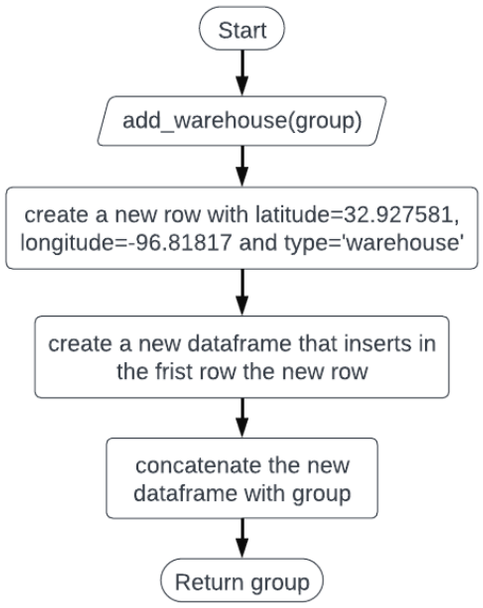


Fig. 10: add_warehouse() flowchart.

Through the utilization of two for loops, '`groupx_w1`' is created given the indexes from '`gx_w1`'. In addition, the function '`add_warehouse`' is implemented to add the information related to the needed warehouse. The following function '`calculate_distance`' calculates the pairwise distances between the several coordinates and converts them into a square matrix that is transformed into a data frame and then a list.

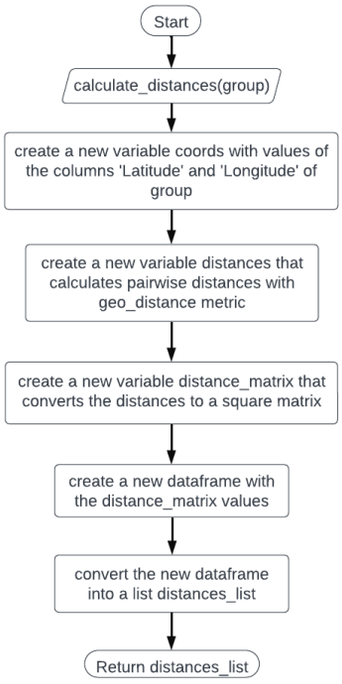


Fig. 11: calculate_distances() flowchart.

Using another for loop, it is possible to append the calculated distances of '`groupx_w1`' in an empty list called '`distance_gx_w1`'. A new column '`Group`' in '`group1`' is then created and filled by the string 'Group' for each row.

Each shipping point is now assigned to a particular subgroup and this is done through the function here below:

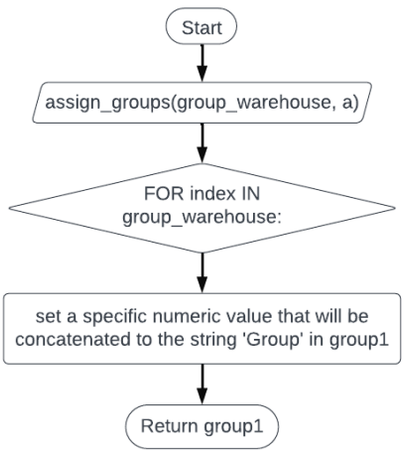


Fig. 12: assign_groups() flowchart.

To increment the value of the number by one each time, another for loop is needed. As a matter of fact, scrolling the indexes from 0 to 9 and initializing a counter that increments itself by one at every loop (given the index), it is possible to assign the correct group number to each shipping point in '`group1`'.

Now all the clusters are illustrated with ten different colors (one per cluster).

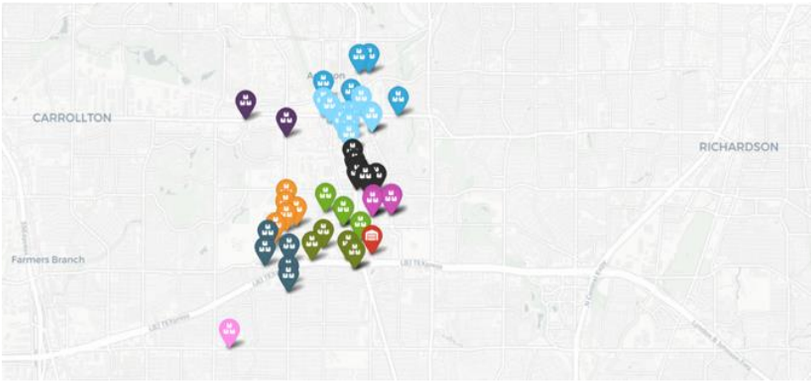


Fig. 13: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

1.3.1 First scenario: VRP solution with OR-Tools.

OR-Tools is an open-source software as mentioned in [16] and it is deployed to optimize the solution for the VRP just proposed. Among all the 10 sub-groups, the sub-group number two is chosen for being solved and the routes are evaluated with the crow fly distance. Firstly, two different dictionaries (one for the longitudes and one for the latitudes) are generated: their keys are the facilities' type (either warehouse or shipping point 0, 1, 2, etc.) and their longitudes and latitudes are encoded in the values of each key, respectively.

Similarly, the demand for each shipping point is established by firstly setting a random seed of 2021. Later, '`demand_g2_w1`' is created by generating a list of random absolute values with mean equal to 1, standard deviation equal to 0.4 and size equal to the length of the second group multiplied by 400 for scaling reasons.

The function '`create_data_model`' takes under consideration also the weight capacity of each truck and the weight of the packages collected.

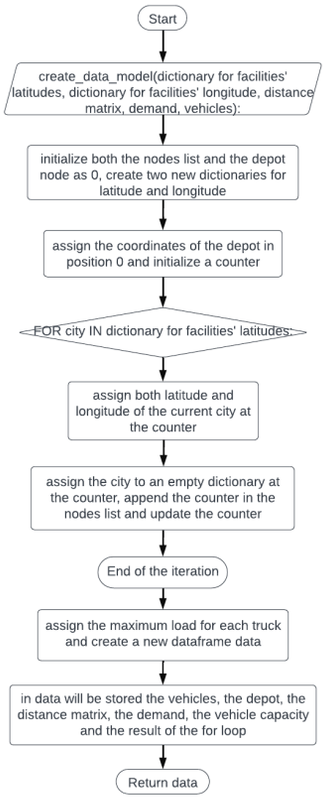


Fig. 14: create_data_model() flowchart.

Thanks to this function, it is possible to store all the necessary data that is going to be used in the function '`main`' provided by the Google OR-Tools documentation [15]. This data is stored in a new variable called `data`, in which the function '`create_data_model`' is called.

Once the data model is created, the function proposed by [15] is implemented and this is the solution that the two executions together give:

```
Route for vehicle 0:
 0 Load(0) ->  1 Load(638) ->  0 Load(638)
Distance of the route: 6km
Load of the route: 638kg

Route for vehicle 1:
 0 Load(0) ->  3 Load(333) ->  2 Load(841) ->  0 Load(841)
Distance of the route: 7km
Load of the route: 841kg

Route for vehicle 2:
 0 Load(0) ->  5 Load(488) ->  4 Load(758) ->  0 Load(758)
Distance of the route: 6km
Load of the route: 758kg

Total distance of all routes: 19km
Total load of all routes: 2237kg
{0: [0, 1, 0], 1: [0, 3, 2, 0], 2: [0, 5, 4, 0]}
<ortools.constraint_solver.pywrapcp.Assignment; proxy of <Swig Object of type
'operations_research::Assignment *' at 0x7fa8ebb315a0> >
```

Each vehicle has a specific route to follow, given the number and which shipping points must serve: this is highly correlated to the crow fly distance that solves the objective function disclosed before. It is noticed that for every single vehicle, both the distance traveled and the load of each route is displayed as well as the total distance and total load of all routes.

At the end, a dictionary is returned:

- Its keys represent the vehicles.
- Its keys' values include a list of the shipping point that each truck must fulfill and in which order, starting and finishing with the depot.

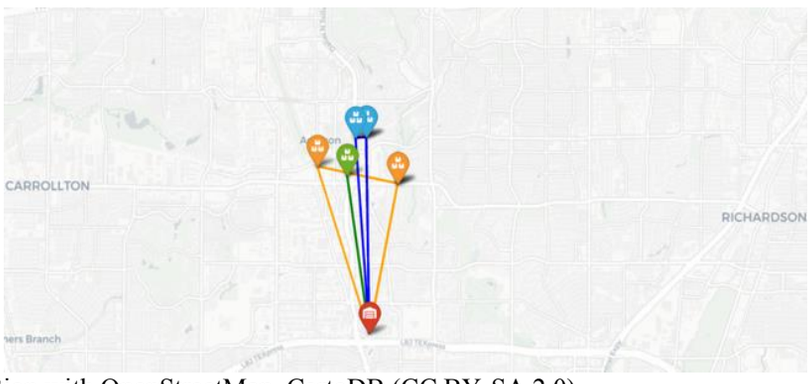


Fig. 15: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

1.3.2 Second scenario: VRP solution with APIs of ORS.

In this second scenario, new constraints are added: indeed, the trucks won't fly on a crow fly distance anymore since they will follow the authentic pathways of the city. What is more, travel time constraints are included as well.

The new parameters are:

- d_i = due time for facility i (06:30 pm).
- r_i = release time of demand of facility i.
- t_{ij} = travel time from facility i to facility j.
- c_{ij} = travel cost from facility i to facility j.

The new decision variables are:

- z_{ik} = arrival time of vehicle k at facility i.
- z_{0k} = departure time of vehicle k from the warehouse.

The new objective function aim is to maximize the amount picked up by each truck:

$$max(X) \sum_{i \in F} \sum_{j \in F} \sum_{k \in K} q_i x_{ijk}$$

The new constraints, instead, are the following:

- Every truck must leave the warehouse only when facility i is open:

$$z_{0k} \geq \sum_{i \in F} x_{ijk} \; \forall i \in F, \forall k \in K$$

- Every truck must complete all the previous required operations before arriving to the next facility:

$$z_{ik} + s_i + t_{ij} \leq z_{jk} \; \forall i, j \in F$$

In the first place, what is needed is a function able to compute the final cost for each vehicle, bearing in mind the distance covered, the average speed and the price per kWh.

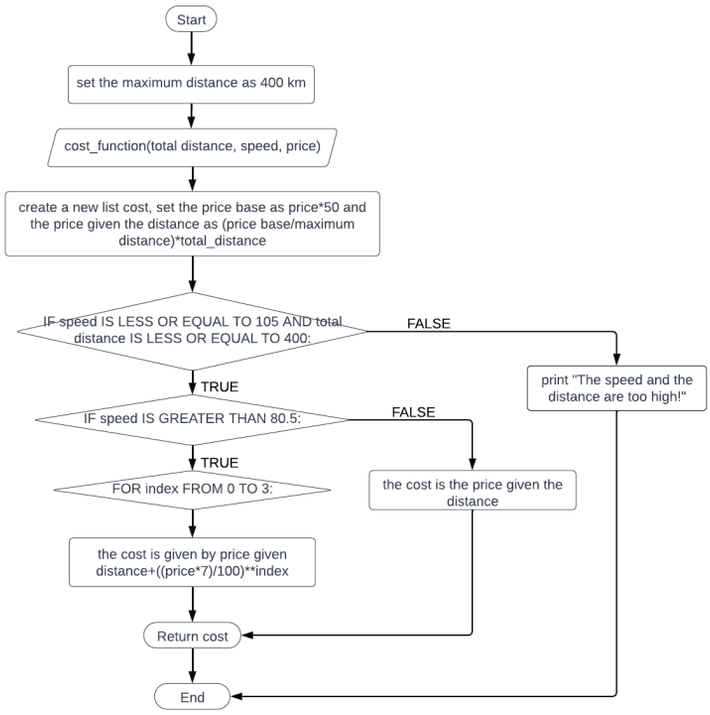


Fig. 16: cost_function() flowchart.

Since the time window constraint has similarly been embedded, a new data frame with updated information about the opening and closing time of the facilities is produced.
Hereinafter, the first rows of this new data frame regarding all the ten groups:

	Latitude	Longitude	Type	Group	Open_from	Open_to	ID	Amount
0	32.955595	-96.823273	shipping point 0	Group 2	2023-04-11 14:30:00	2023-04-11 18:30:00	0	518
1	32.957150	-96.829808	shipping point 1	Group 2	2023-04-11 14:30:00	2023-04-11 18:30:00	1	412
2	32.954141	-96.811909	shipping point 2	Group 2	2023-04-11 14:30:00	2023-04-11 18:30:00	2	270
3	32.950938	-96.818244	shipping point 3	Group 7	2023-04-11 14:30:00	2023-04-11 18:30:00	3	220
4	32.951139	-96.818345	shipping point 4	Group 7	2023-04-11 14:30:00	2023-04-11 18:30:00	4	397
5	32.939568	-96.817191	shipping point 5	Group 10	2023-04-11 14:30:00	2023-04-11 18:30:00	5	233
6	32.939414	-96.819821	shipping point 6	Group 10	2023-04-11 14:30:00	2023-04-11 18:30:00	6	472
7	32.935293	-96.817894	shipping point 7	Group 4	2023-04-11 14:30:00	2023-04-11 18:30:00	7	408
8	32.935055	-96.817898	shipping point 8	Group 4	2023-04-11 14:30:00	2023-04-11 18:30:00	8	338
9	32.940934	-96.821734	shipping point 9	Group 10	2023-04-11 14:30:00	2023-04-11 18:30:00	9	379
10	32.940656	-96.821736	shipping point 10	Group 10	2023-04-11 14:30:00	2023-04-11 18:30:00	10	341
11	32.947361	-96.823642	shipping point 11	Group 7	2023-04-11 14:30:00	2023-04-11 18:30:00	11	216
12	32.947418	-96.823642	shipping point 12	Group 7	2023-04-11 14:30:00	2023-04-11 18:30:00	12	378
13	32.934817	-96.818099	shipping point 13	Group 4	2023-04-11 14:30:00	2023-04-11 18:30:00	13	338
14	32.940017	-96.821739	shipping point 14	Group 10	2023-04-11 14:30:00	2023-04-11 18:30:00	14	76
15	32.944085	-96.822906	shipping point 15	Group 10	2023-04-11 14:30:00	2023-04-11 18:30:00	15	468

Table 1: part of the data frame containing information about the facilities of all the ten different groups.

The first two columns contain the latitude and longitude of each facility, while the other columns enclose the type of the facilities, their pertaining group, opening hours, IDs and the demand amount respectively. A few considerations are made: the solution is found for a given day (April 11th, 2023) and the demand value of the customers is different since it is generated for all the groups of the first warehouse.

In this second scenario, the analysis is based on Group 2.

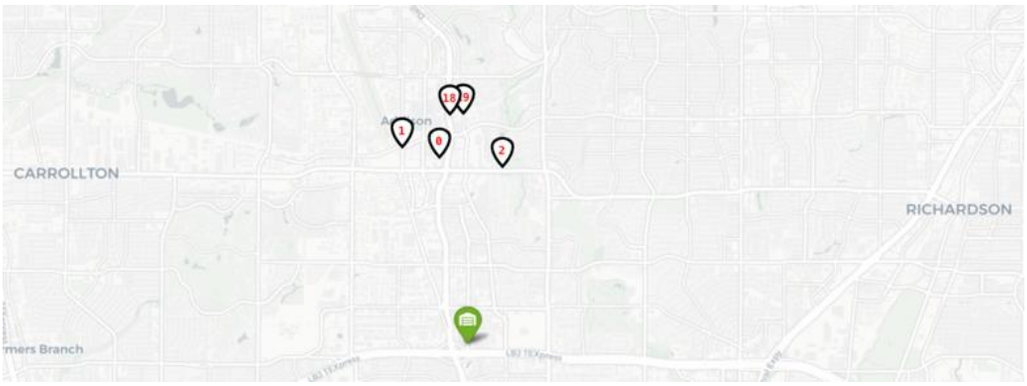


Fig. 17: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

The solution provided by OR-Tools is here described:

```
Route for vehicle 0:
 0 Load(0) ->  3 Load(270) ->  0 Load(270)
Distance of the route: 6km
Load of the route: 270kg

Route for vehicle 1:
 0 Load(0) ->  1 Load(518) ->  0 Load(518)
Distance of the route: 6km
Load of the route: 518kg

Route for vehicle 2:
 0 Load(0) ->  5 Load(364) ->  4 Load(515) ->  2 Load(927) ->  0
Load(927)
Distance of the route: 6km
Load of the route: 927kg

Total distance of all routes: 18km
Total load of all routes: 1715kg
{0: [0, 3, 0], 1: [0, 1, 0], 2: [0, 5, 4, 2, 0]}
<ortools.constraint_solver.pywrapcp.Assignment; proxy of <Swig Object of type
'operations_research::Assignment *' at 0x7fa8deda3cc0> >
```

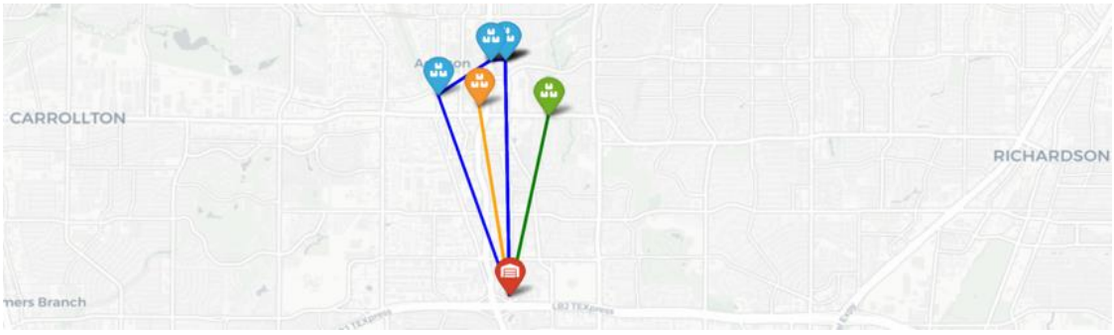


Fig. 18: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

OpenRouteService (ORS) [16] is an open-source routing service that offers the possibility to avoid crow fly ranges and to compute the distances among the routes in concrete terms. This is achievable

by means of Application Programming Interface (API) keys [17], which help in the identification and validation of either a user or an application.

The first step is to develop two functions:

- One defines the vehicles implemented in the resolution of the problem. As a matter of fact, with the '`ors.optimization.Vehicle`' built-in function, a vehicle with a single ID, specific starting and ending locations, capacity and time windows in POSIX [18] format, is created.

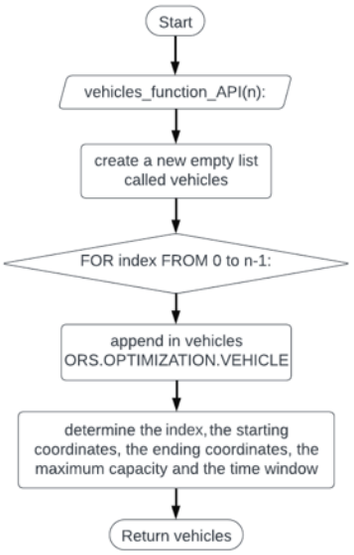


Fig. 19: vehicles_function_API() flowchart.

After defining the function, a new variable `vehicles` is set recalling '`vehicles_function_API`' with n equal to 3 (which is the number of vehicles).

- The second one lays down the delivery stations and takes the needed information from the Group 2 dataset. With the '`ors.optimization.Job`' a task is being represented, which in this case involves the delivery of the packages from the shipping points to the final destination. Each job has a unique ID, the location of every single pickup, the service time required to accomplish the duty (40 minutes), the amount associated and time windows in Unix timestamp.

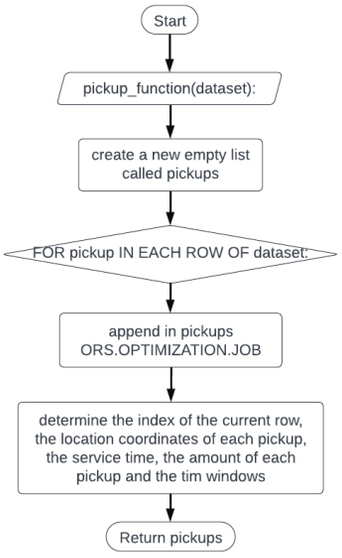


Fig. 20: pickup_function() flowchart.

After defining the function, a new variable `pickups` is set recalling '`pickup_function`', taking as a dataset the one containing all the information related to Group 2 (in the project it is called '`pickups_data`').

Now, a client can be initialized and a request is made.

```
ors_client <-- ORS.CLIENT(API KEY)
result <-- ORS_CLIENT.OPTIMIZATION(
     jobs <-- pickups
     vehicles <-- vehicles
     geometry <-- True)
```

It is found that only two trucks are necessary for this type of solution and not three as seen before. This is because trucks are following real routes which enables them to optimize both the traveled time and distance.

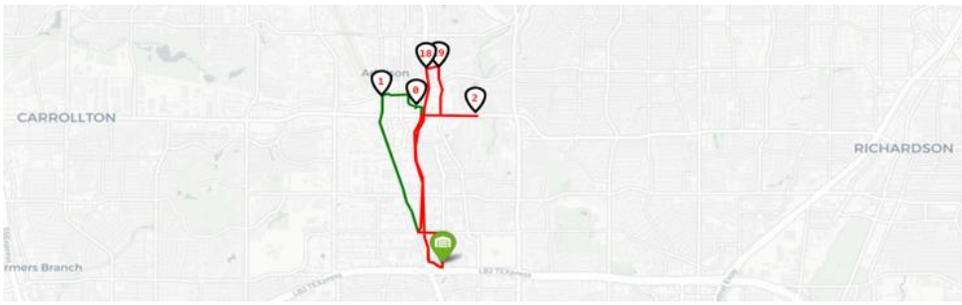


Fig. 21: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Now, the transportation cost for each vehicle is computed by using the speed, the distance and the price of the electric fuel. To get this result, a new data frame '`vehicles_df`' is created: this contains the distance in meters, the amount in kilos and the travel duration in seconds for each vehicle. Subsequently, the distance is transformed in kilometers, the duration in hours and the speed (km/hours) is added.

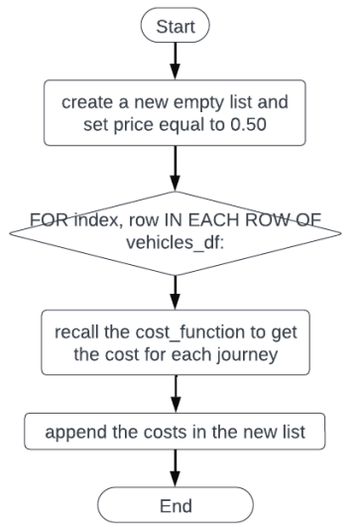


Fig. 22: for loop cost flowchart.

The cost column is added to the '`vehicles_df`' data frame, showing that vehicle 1 had to face a higher variable cost rather than vehicle 0.

vehicle	distance	amount	duration	time	speed	cost
0	8.59	[930]	0.24	861	35.79	0.54
1	10.73	[785]	0.30	1075	35.77	0.67

Table 2: information of each used vehicle.

Furthermore, the vehicles' schedule is another important aspect for the resolution of this VRP. With the following piece of code, creating a data frame for every vehicle that specifies the arrival and departure time for all the shipping points and the depot, is achievable. Of course, the 40 minutes of service time employed to load the packages are taken under consideration.

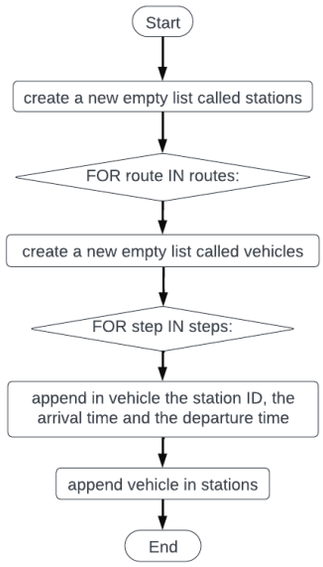


Fig. 23: for loop vehicles' schedule flowchart.

	Station ID	Arrival	Departure
0	Depot	2023-04-11 14:24:21	2023-04-11 14:24:21
1	1	2023-04-11 14:30:00	2023-04-11 15:10:00
2	0	2023-04-11 15:12:13	2023-04-11 15:52:13
3	Depot	2023-04-11 15:58:42	2023-04-11 15:58:42

Table 3: schedule of vehicle 0.

The rows follow the graphical scheme above reported, starting from the depot and going to all the shipping points indicated. The shift should end at 06:30 pm and it is noticeable that every truck finishes before that binding schedule.

	Station ID	Arrival	Departure
0	Depot	2023-04-11 14:24:17	2023-04-11 14:24:17
1	18	2023-04-11 14:30:00	2023-04-11 15:10:00
2	19	2023-04-11 15:10:56	2023-04-11 15:50:56
3	2	2023-04-11 15:54:01	2023-04-11 16:34:01
4	Depot	2023-04-11 16:42:12	2023-04-11 16:42:12

Table 4: schedule of vehicle 1.

1.3.3 Comparison between the two scenarios.

In this section of the first chapter, two summarizing tables reporting the pros and the cons of the scenarios just explained, are shown [19][20]:

OR-TOOLS PROS	OR-TOOLS CONS
Open-source optimization solver.	Optimal solution difficult to choose.
Quick path results, very low costs and easy to use.	Not applicable to real-world scenarios and restricted available documentation.
Constraints control.	Constraints in the `RoutingModel` class are the only supported.
Used for packing, scheduling and routing.	Difficult to customize.
Fast bug resolution and usual upgrades.	In this case, only crow-fly distance solutions.

Table 5: pros and cons of scenario 1.

ORS PROS	ORS CONS
Open-source, updated tool and more affordable than many others.	No available data about neither the traffic nor the live traffic.
A lot of available information about the routes.	Limited geocoding scope.
Flexible tool able to customize the model.	Sporadic service suspension.
APIs enable integration.	
In this case, also real distance solutions.	

Table 6: pros and cons of scenario 2.

1.4 Literature analysis.

The literature analysis for this topic aspires to offer to the reader an exhaustive perception of the state-of-the-art related to the VRP – more specifically VRPTW – solved through heuristic methods. With the help of available research papers and existing approaches, trends and patterns can be easily identified and the review based on the work done until now is straightforward.

By means of this revision, it is possible to provide a stable theoretical foundation determined also by the academic notions gathered in the first paragraphs of this chapter. Significant frameworks for the research are shown and the methodologies exploited to solve the FedEx VRPTW, demonstrated.

As already previously affirmed, heuristic (non-exact) algorithms can be divided in three main categories: constructive, improvement (classical) and metaheuristics. Generally, their solution is said to be "*satisfactory*" at a "*reasonable computational cost*" [7]. Building a solution from scratch means creating a route-building heuristics, while route-improving heuristics try to refine an already existing solution: this is the main difference between constructive and improvement algorithms.

A clear example of constructive heuristic is the Clarke and Wright (1964) algorithm based on VRP: this method envisages the fulfillment of the task by supplying every client with a different truck. If any two of the single available routes were combined, then the cost would be reduced as one less truck would suffice. The cost of satisfying customer i and j by two distinct vehicles is $c_{0i} + c_{i0} + c_{0j} + c_{j0}$ when the same cost but using a single vehicle, would be $c_{0i} + c_{ij} + c_{j0}$, deriving the following cost saving $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ [21].

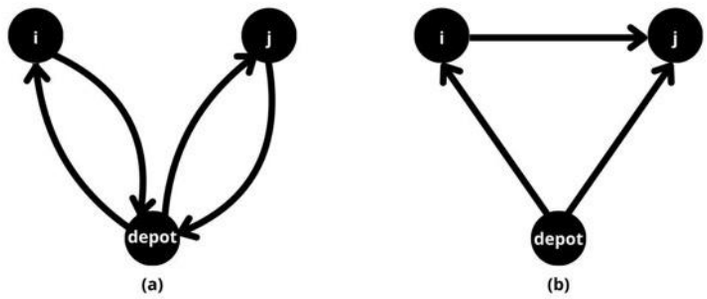


Fig. 24: S. Halim, L. Yoanita, Adjusted clustering Clarke-Wright Saving Algorithm for two depots-N vehicles, *Semantic Scholar*, December 1st, 2015. [Online]. Available: https://www.semanticscholar.org/paper/Adjusted-clustering-Clarke-Wright-Saving-Algorithm-Halim-Yoanita/d921cd73380eed72c69ae4b34523d2fb8cfbebad (recreated image).

The real first published study on VRPTW has been conducted by Baker and Schaffer in 1989 (an expansion of the Clark and Wright algorithm): their algorithm (based on a saving heuristic) starts with all the feasible routes starting from the depot, getting to a single customer and coming back to the depot (depot-*i*-depot). To find an attainable resolution, a set of two routes is created at each recurrence maximizing the saving between customers i and j, that would be $s_{ij} = t_{i0} + t_{0j} - Gt_{ij}, i \neq j, i, j = 1,2, \dots, n$ where G can be expressed as the "*route form factor*" [22].

The route-improvement heuristics work on already existing routes and base its theories on the concept of neighborhood as it mainly employs the local search algorithm, as defined in [22]. By iteratively investigating the neighborhood of a certain solution and by carrying out small-scale adjustments time by time, they are able to shift to better solutions. For instance, Christofides and Beasley (1984) exploited the k-node interchange operator as described in [22], but other algorithms leverage more operators such as relocate, exchange, 2-Opt∗, etcetera.

A frequent challenge in improvement heuristics is to overcome local optima, which may not be the global optimal solution: they basically embody points for which it is not possible to get additional amelioration with local search algorithms. For this reason, metaheuristic methods and other diversification strategies are used to look for better solutions.

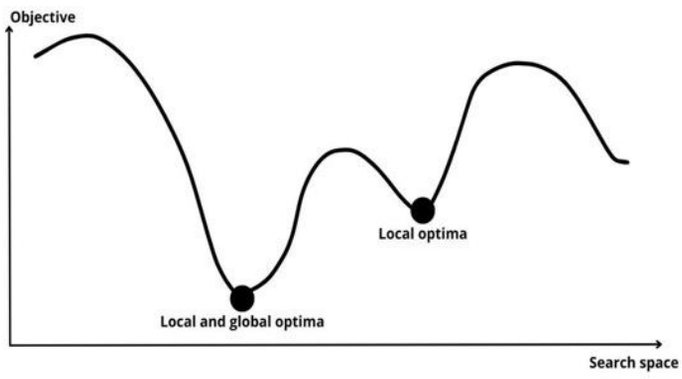


Fig. 25: Bill, R., Fleck, M., Troya, J. et al. A local and global tour on MOMoT, Softw Syst Model 18, 1017–1046 (2019). [Online]. Available: https://doi.org/10.1007/s10270-017-0644-3 (recreated image).

Techniques as genetic algorithms, tabu search and ant colony optimization are included in the metaheuristic category that is defined in [24]. They have the power to operate on already existing solutions – either complete or defective – and therefore, improve them. Additionally, this type of heuristics can be split into two separate classes: single-solution-based (local search-based) and population-based methods. As pointed out by [25] they have the common characteristic to perform a

low-level search through the implementation of improvement heuristics. High-level executions will instead try to find a remedy for the local optimum problem.

One of the oldest single-solution-based methods is the tabu search, first established around 30 years ago. Tabu search assesses a set of neighbors of the current solution in each iteration, keeping the best one it found to surmount local optima [26]. Even if it seems to be flexible and efficient in the way it operates and it can provide actual results, no research has been capable of demonstrating its convergence so far and [22] suggests an alternative way to improve it.

Population-based methods ground their theories on natural notions including the evolution of species and the attitude of insects in their society (e.g., ants) and make use of high-level methods like Neural Networks (NN). In the context of VRP, literature shows that hybrid approaches are mostly executed as a first local search is usually needed. Ant Colony Optimization (ACO) and genetic algorithms are vivid examples of these techniques and they are delineated [25].

Under a different circumstance from the one carried out throughout this thesis, as the one of the Vehicle Problem with Simultaneous Delivery and Pickup (VRPSDP), [27] tried to develop the Artificial Bee Colony (ABC) algorithm minimizing the cost of transport vehicles knowing that distribution activities are executed at the same time. This type of computation (metaheuristic) aims to optimize parameters, basing the study on the bees' society and their behavior, dividing them in three categories (scout, employed and onlooker bees). Here the ABC algorithm constructed for the VRPSDP:

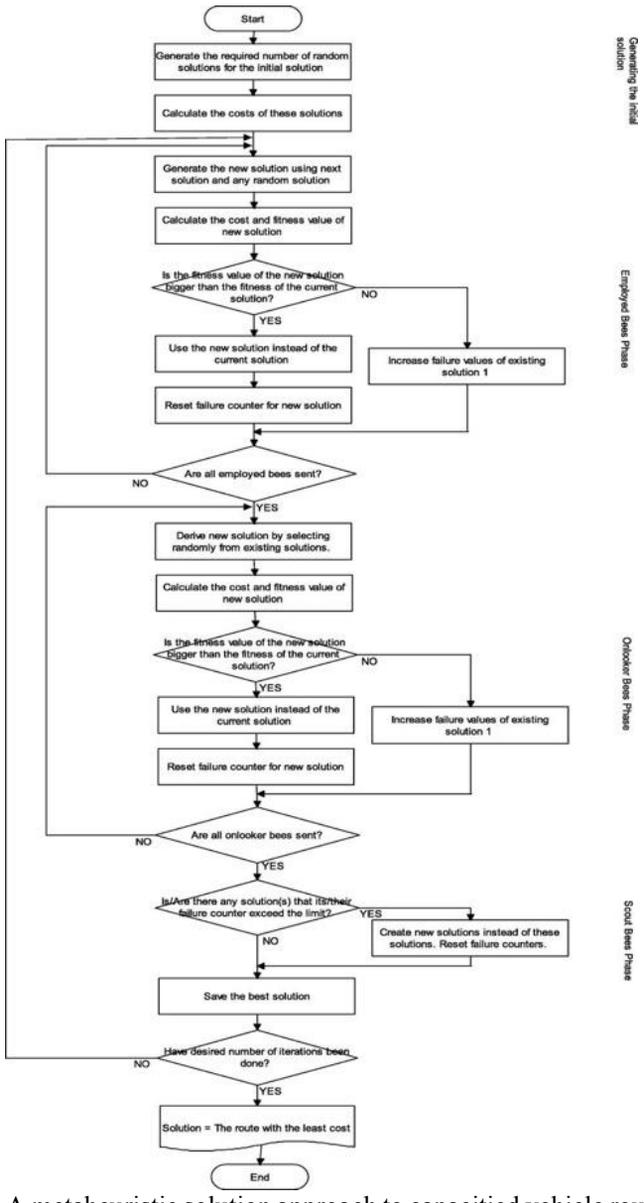


Fig. 26: F. Simsir, D. Ekmekci, A metaheuristic solution approach to capacitied vehicle routing and network optimization, *Sciencedirect*, January 23rd, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098618320962

Approaching hybrid heuristics, [28] proposes a VRPTW with compatibility constraints in the context of Home Healthcare (HHC) system, regulating – and scheduling – the relationships between the patient (customer) and the caretaker (server, provider) within a precise time frame. To produce initial solutions, Particle Swarm Optimization (PSO) and the inserting procedure are combined; local search is said to improve those solutions; for the global search, path relinking is implemented; if the algorithm is stuck in the local optimum, then the decision makers will make use of the ruin-and-recreate (R&R) procedure.

Therefore, it is possible to understand new emerging trends for VRP and VRPTW:

- Hybrid approaches are becoming increasingly used as they improve the quality of the solution and the computational efficiency.
- Adaptive and self-adaptive metaheuristic methods boost the versatility of the algorithm, as developed in [29] and [30].

On the other hand, heuristics for large-scale VRPs have not reached the optimal resolution yet, as they still need to face important challenges. The most important ones are listed in [31]:

- The generalization from small-scale VRPs to large-scale VRPs is complex and it usually requires retraining.
- Real-time solutions are still a delicate topic.
- Global constraints might encounter obstacles when being incorporated in learned heuristics.

[29] also suggests some ways to overcome these gaps and challenges, as for example the execution of a Two-stage Divide Method (TAM) that embeds a two-step RL technique to train it.

1.5 Conclusion.

This chapter broke down a VRP structure with the time window constraint, which has been solved through the utilization of greedy methods in two different scenarios (OR-Tools and ORS resolution). It also analyzed the available literature that proved the success of heuristic methods when dealing with optimization problems.

Thanks to this preliminary chapter, it is going to be easier to solve a VRPTW for the same FedEx dataset through the enforcement of RL. As a matter of fact, RL can be a real tool able to optimize and refine greedy techniques, getting more accurate solutions and surmounting the challenges that heuristics can't technically overcome.

In the following chapter of the thesis, a resolution proposal with RL is going to be given. Pseudocode and description of the latter are granted as in the last paragraphs, to better lead the reader to the comprehension of the multiple adjustments.

Chapter 2. Resolution proposal of the optimization problem with RL.

2.1 Introduction.

Reinforcement Learning is "*the science of decision making*" and it can exploit the environment around itself to learn the "*optimal behavior*" and to finally be rewarded with the highest prize [32]. RL algorithms are categorized in the ML sphere (together with Supervised and Unsupervised Learning), which is again a branch of Artificial Intelligence (AI) as can be seen in the following scheme.

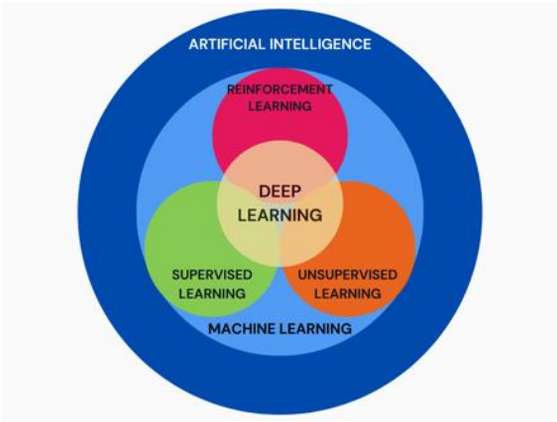


Fig. 27: A. Karthikeyan, Artificial intelligence: machine learning for chemical sciences, *Researchgate*, September 14th, 2022. [Online]. Available: https://www.researchgate.net/publication/349058264_Machine_Learning_methods_for_solving_Vehicle_Routing_Problems (recreated image).

[33] offers a concise example that explains what RL really is:

> "*Imagine a robot whose body includes a standing component that must balance on top of a cart with wheels. The robot is the agent and can take the actions of moving left and right in its environment. When the robot balances successfully it receives a reward.*"

In addition, this robot is able to come up with four categories of observations which include the velocity and the position of the cart, the robot body's angle and velocity. To estimate the probability of unexpected events, it's assumed that the agent makes its own decisions within the boundaries of the Markov Decision Processes (MDPs). [34] provides a detailed explanation about MDP and the most relevant variables of this framework, which are:

- S = states.
- A = actions.
- $P\ (St + 1|st.at)$ = transition probabilities.
- $R\ (s)$ = reward.

The MDP model can be represented as in the below graph:

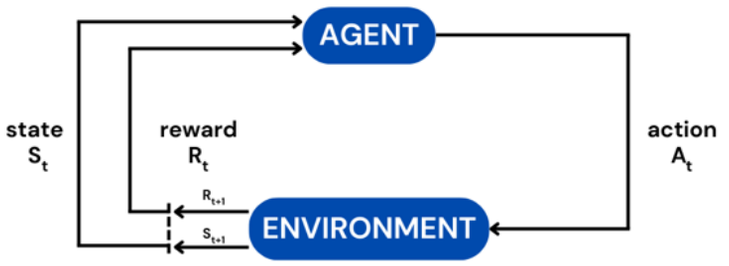


Fig. 28: Vijay Kanade, What Is the Markov Decision Process? Definition, Working, and Examples, Spiceworks, December 20th, 2022. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-markov-decision-process/ (recreated image).

The Markov Property must respect the subsequent equation as widely discussed in [34]:

$$P[St + 1|St] = P[St + 1|S1, S2, S3, \dots, St]$$

This type of structure works well in the field of optimization routing problems. To tell the truth, a problem such the VRP can be decomposed like this:

- The agent corresponds to the vehicle (or set of vehicles) implemented in the resolution.
- The actions are represented by the available paths and routes that the trucks can take.
- The rewards symbolize the costs to bear if the DM opts for a specific route to follow or for a particular action.
- The goal instead, is to achieve an optimal policy able to minimize the costs of the objective function.

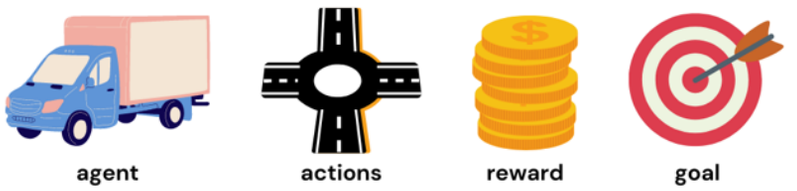


Fig. 29: The elements of the MDP model for a VRP/VRPTW.

In RL, an extensively used algorithm is the one called Q-Learning and [35] offers a meticulous interpretation of this computation.

Here below, the flowchart of the Q-Learning algorithm.

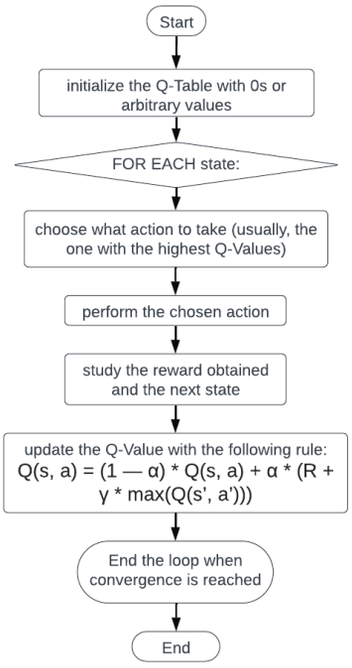


Fig. 30: Q-Learning flowchart from [35].

2.1.2 Uncertainties of the environment and competitive methods.

When dealing with a company like FedEx that guarantees a delivery service to its clients, a lot of uncertainties can alter the business. For instance, some external providers could face unexpected delays or the demand could be facing either low or high levels. Moreover, the travel time of the trucks can be dictated by several features that can't be controlled by the company (e.g., traffic, temperatures and weather conditions). Therefore, in this new implementation, it is essential to consider uncertain factors that will probably influence the resolution of the VRP.

In line with [36], "*a stochastic process or system is connected with random probability*." Even if a lot of stochastic VRP approaches presume normal distribution for the travel time and the demand, in the real world they follow an unknown distribution. As a result, the main goal is to train on historical data as specified by [37].

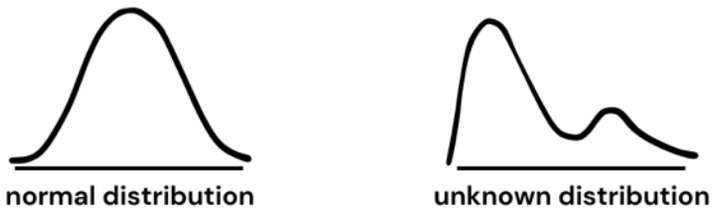


Fig. 31: The normal and the unknown distribution patterns.

The development of a *competitive model* is indeed essential to find feasible solutions to the VRP/VRPTW so, – according to [37] – it must:

- Be able to generalize to unseen problems.
- Be safe.
- Be faster than the heuristic methods.
- Create almost optimal solutions.

2.1.3 Literature analysis.

A quite recent study carried out by M. Nazari (and others) [38], proposes an end-to-end framework to resolve the VRP using RL. In contrast with the type of VRP described so far, this one provides for picking up the necessary goods at a given depot and the subsequent delivery to the multiple client locations.

Here, the agents (the fleet of vehicles) understand the interrelationships with the environment and get a positive reward every time they fulfill their task (successful shipment).

The authors' goal was to train only one model for a specific type of instance able to solve any other similar instance. As a matter of fact, they trained a solver (a black box) by updating its parameters every time with the view of increasing the probability of good routes and the contrary for bad routes. This process can be accomplished thanks to their model that employs a Recurrent Neural Network (RNN) to develop a probability distribution [37] [38].

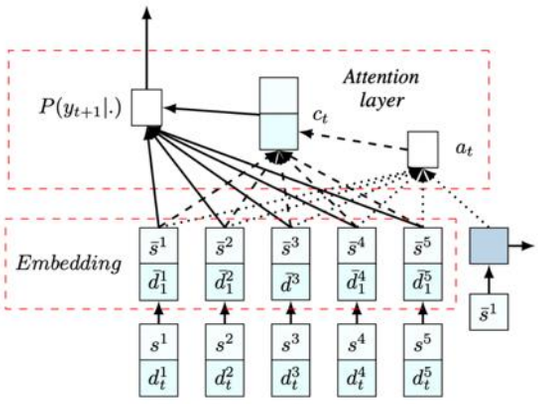


Fig. 32: Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, Martin Takáč, Reinforcement Learning for Solving the Vehicle Routing Problem, arXiv, February 12th, 2018. [Online]. Available: https://arxiv.org/abs/1802.04240.

According to the results they achieved, RL can improve the outcomes (with respect to greedy techniques, including OR-Tools) as it can provide near-optimal solutions.

Furthermore, a new research paper investigated how to optimize the travel costs of an SVRP (Stochastic Vehicle Routing Problem) with Time Windows. This means that the model involves unknown parameters (as defined in 2.1.2) for instance demand and travel costs [39]. These unknown parameters include stochastic and external variables that may depend on the type of service or good the company deals with.

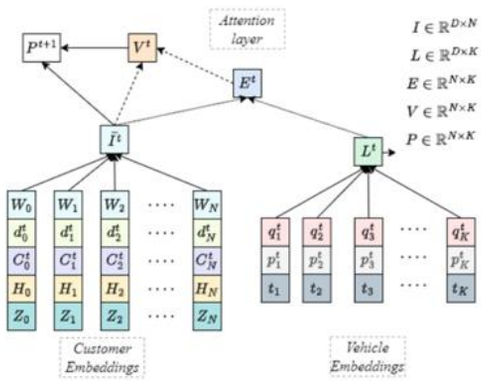


Fig. 33: Z. Iklassov, I. Sobirov, R. Solozabal, M. Takáč, Reinforcement Learning for Solving Vehicle Routing Problem with Time Windows, arXiv, February 15th, 2024. [Online]. Available: https://arxiv.org/abs/2402.09765

The result obtained is an improvement of 1.73% with respect to the metaheuristics, which leads to a minimization of the travel costs. In addition, as reported in [39], the model seems to easily adapt to different variables and circumstances and consequently, to find a proper solution. On the other hand, when addressing the issue to more than one vehicle (therefore, creating a bigger action space) the improvement decreases, resulting in decreasing returns after the optimal number of vehicles provided.

As a matter of fact, some limitations and risks associated to the use of RL for VRP might be [40]:

- Large amount of historical data is needed and its quality must be ensured.
- The reward function must not be too complex, as it may restrict the productiveness of the model.
- Greedy techniques have the capacity of generalizing better than RL models.
- Sometimes, RL are challenging to interpret provoking the Black Box problem.

2.2 Solution proposal with RL.

In this section, the main goal of this thesis is accomplished: providing a resolution of the VRP/VRPTW through RL, according to its laws and modalities.

2.2.1 Problem statement.

The problem that is going to be solved reflects the one described in Chapter 1. More precisely, it covers the purposes highlighted in 1.3.2, as RL takes also under consideration additional constraints (e.g., time windows). As the second scenario remarks, the objective is to maximize the amount picked up by each truck; in addition, the following resolution wants to minimize the total distance covered. This means that the objective function will be the minimization of the costs that FedEx must face.

The simulation environment in which the agent will run [41], is initialized with the most relevant parameters such as the number of customers, the number of vehicles and their capacity and the time windows. Its key components are represented by the state, the action, the reward and the policy and each one of them is going to be fully described for this specific case.

As for the state, it is portrayed by the current location of the vehicle and the demand of each customer. These two elements help the agent to make the right decisions as they provide valuable details for the problem to be solved [42]. With reference to the action and the observation spaces, the following can be said: the action space is discrete (an action entails choosing which customer is going to be visited next), while the observation space is continuous (ranging from 0 to 1).

One of the methods included in the environment regards the rewards conferred to the agent which will update its policy and boost the decision-making process. The assignment of these rewards is explained throughout the chapter.

In addition, both external and stochastic variables can be added so as to capture the reality of the case. As a matter of fact, the agent is not able to control all the conditions of the problem and these include traffic situations and road closures among others. External variables have a huge impact on the environment and for this reason, they need to be taken into consideration. Uncertainty is conversely established by stochastic variables which portray probabilistic events such as delays or eventual interruptions of the daily job.

At the same time, the policy serves as the force that shapes how trucks move among the several pick-up points, considering all the constraints (including the possibility of getting either a reward or a penalty). The learning method [43] can be *value-based*, when the policy is not explicitly indicated

(*policy-based*); a value function is then used and, in order to maximize the expected cumulative rewards, Q-Learning or Deep Q-Networks (DQN) can be implemented.

With the purpose of evaluating the performance of the agent, learning curves can be realized. This allows them to supervise the progress of the agent and recognize possible problems that may emerge [44]. For instance, cumulative or average rewards can be plotted as the success rate in the long run. [45] provides a detailed explanation of the several "*key metrics*" used the most in this field.

2.2.2 Models implemented and hyperparameter tuning.

The models considered for this specific study are called A2C (Advantage Actor Critic) and PPO (Proximal Policy Optimization).

A2C (deterministic) takes part of a group of "*hybrid-approach*" algorithms [46], because it can train the agent based both on policy-based methods (Actor) and on value-based methods (Critic). It makes use of an advantage function that helps the DM understand which action is better to take with respect to the average action in a particular state. This algorithm is the major focus of this research. On the other hand, PPO [47] is a policy gradient method that concerns its enforcement on the maximization of the expected cumulative rewards. It serves as a complex neural network structure composed of a policy and a value network, activation and loss functions and an optimization algorithm.

These algorithms won't return successful results if the wrong parameters for the models are chosen. For this reason, it is crucial to perform an hyperparameter tuning every time a change is brought to the environment. There are multiple ways to solve this peculiar activity: a grid search, a random search or even novel methods like Optuna library or the HOOF (Hyperparameter Optimisation on the Fly) [48] method.

First of all, an environment containing all the needed functions should be well described. Then, the environment should be accessed and the researcher be able to interact with it. To do so, it's enough to store all the data in the environment and instantiate it. At each episode, the environment needs to be initialized and that's why the method reset is recalled again at this point. Therefore, it is possible to study the rewards, the penalties assigned to the agent and the route it decides to take before being trained. This will make the reader understand if the balance rewards/penalties is respected and if the agent enjoys a strong foundation with the purpose of learning how to complete the job. Subsequently, the parameters can be tuned (after choosing the algorithm) and used to get better results.

2.2.3 Development of the solution.

Before developing the RL solution, it is important to clarify the contribution of this thesis' author. The helpful resources found online gave the ground basis on which to develop the code and these include the documentation of the libraries Stable Baselines3 [49][50] and gym [51] and the available Github repository of @zestyraiden [52]. Professor Denis Beker provided a wide support for the code to be used for running A2C and PPO models. In addition, ChatGPT helped in the resolution of the errors when writing the code; it was also an advantageous tool that allowed to better understand the outcomes of the research and proposed additional aspects to consider (the use of TensorBoard, for example). The author retrieved the necessary data, developed the logics behind the code (especially the ones reported in the environment) and interpreted the results of the models.

After installing the necessary packages and importing the respective libraries, the `VRPEnvironment` class is developed to allow the agent to learn what it must do. The first function inside the class object is called `__init__()` and it initializes the input parameters

(`num_customers`, `num_vehicles`, `demand_values`, etc.), the attributes (`self.num_customers`, `self.num_vehicles`, `self.demand_values`, etc.) and the methods (the other functions) of the environment.

The development of the solution is split in more scenarios, from the less complicated to the more sophisticated this study could achieve. As a matter of fact, the first scenario doesn't involve the implementation of any hyperparameter or hyperparameter tuning; in addition, it doesn't embrace the "*one customer – one truck*" logic. Instead, it allows all the trucks to travel to an already visited shipping point; this is done to better understand the functioning of the class and the Reinforcement Learning requirements.

Going back to the `VRPEnvironment` class, in the first case also other functions have been created:

- The function `reset()` readjusts the environment to the initial state when a new episode needs to start running.
- The function `load_vehicle()` assigns the logic of the capacity to the trucks. If a given customer's load is greater than the remaining capacity of a truck, the load will be assigned to the subsequent truck (which will visit the customer even in the case where the first truck had already visited it). If the last truck doesn't have enough capacity left, a new truck that will complete the task will be added.

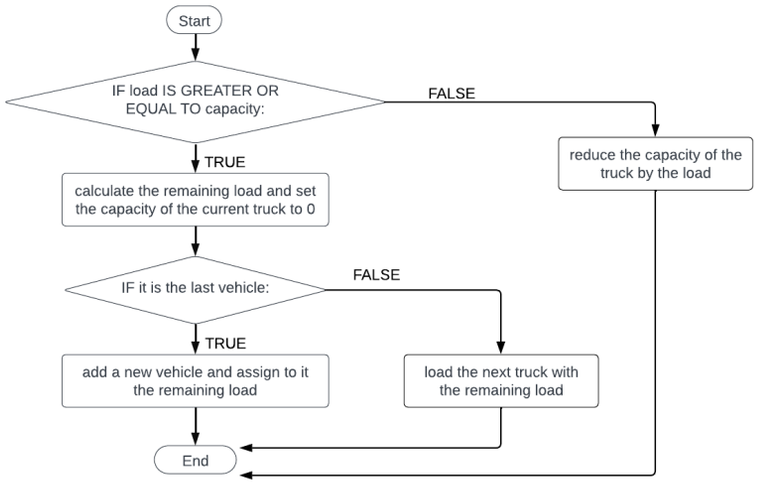


Fig. 34: `load_vehicles()` flowchart.

- The function `add_vehicle()` adds a new vehicle to the environment.

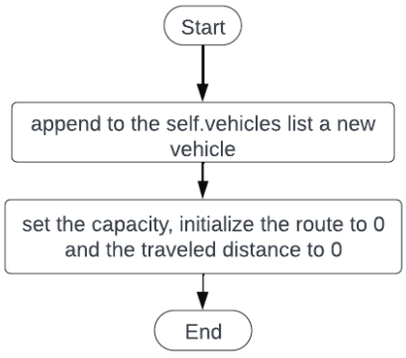


Fig. 35: `add_vehicles()` flowchart.

- The function `step()` performs an action in the environment, updates the state and returns the rewards.

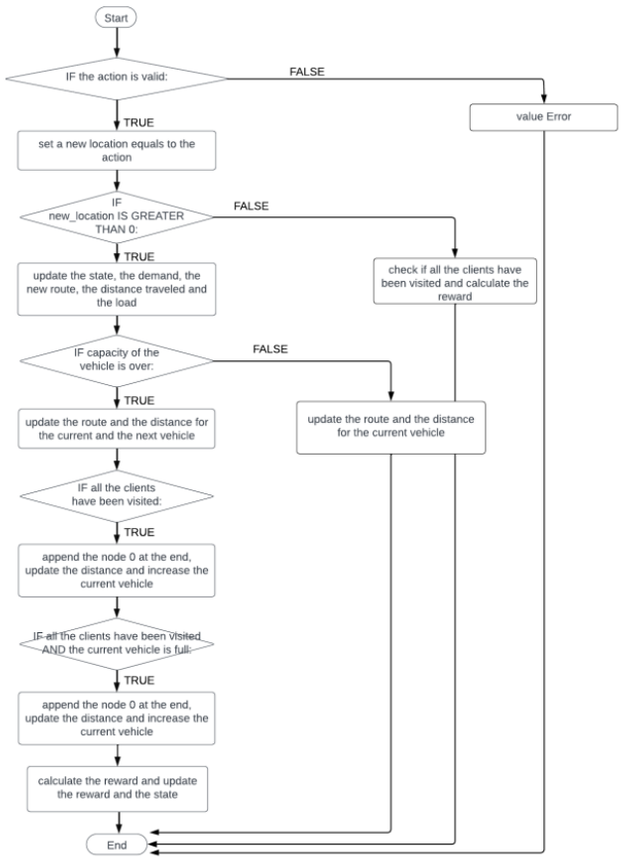


Fig. 36: `step()` flowchart.

- The function `calculate_reward()` allocates the penalties and the rewards to the agent with respect to the deeds performed. In this scenario, the agent receives a penalty for the total traveled distance and receives a reward whenever an episode is completed.
- The function `calculate_distances()` calculates the distance matrix from the latitudes and longitudes of each customer and the given warehouse.
- The function `create_graph()` returns a graph from the distance matrix retrieved from the data explored. Every node contains the weights of the demand of each shipping point, while all the arches describe the distances from a given customer to another one.
- The function `get_routes()` retrieves from `self.vehicles` the path followed by each truck (for example [0, 1, 2, 3, 0], meaning that the truck left the warehouse to visit respectively the customers 1, 2 and 3 and then came back to the warehouse).
- The function `plot_routes_folium()` graphically represents the routes of the trucks in a folium map.

Before training the agent, the environment can be inspected on a single episode by assigning the available FedEx data to the input parameters of the class. From here, it can be seen if the agent is understanding the distinct actions and if the rewards or the penalties are correctly assigned. The parameters are set in this way:

- '`num_customers`' = 5.
- '`num_vehicles`' = 2.
- '`vehicle_capacity`' = 1000.
- '`demand_values`' = '`demand_g2_w1`' (from the previous chapter).

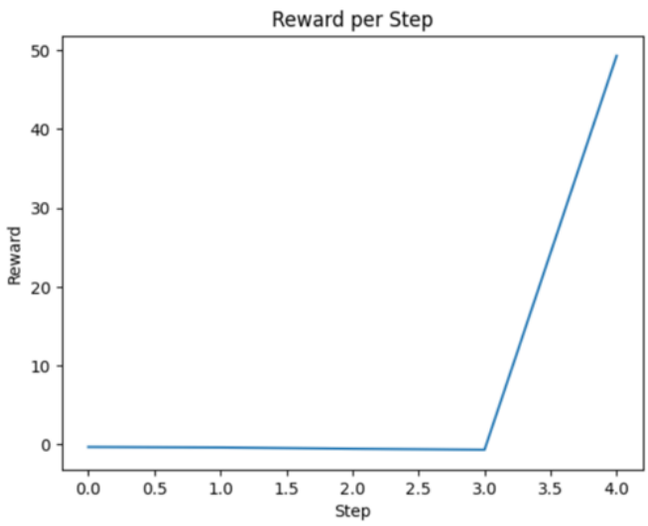


Fig. 37: episode rewards of the environment before training.

In the above graph, the rewards for each step are illustrated. During the first 4 actions, the reward is negative as the agent only earns a penalty; alternatively, in the last action the agent wins a reward of 50 for completing the episode.

Moreover, it is possible to visualize the routes covered by the trucks:

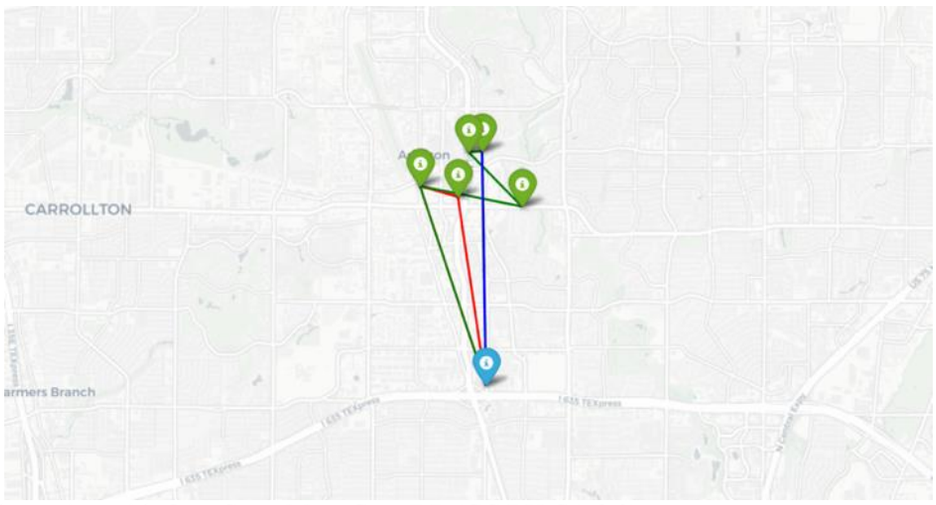


Fig. 38: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

At first sight, it can be noticed that the solution is not optimal: the trucks are wasting too much time and too many resources to complete the task. In fact, it looks like they are visiting the customers randomly instead of following a precise order. The enhancement of the routes would focus on allowing the trucks to create ordered paths in which they can visit the clustered shipping points, according to the minimum distance between one customer and the other. This will save time and the vehicles' fuel.

The first truck (red line) is visiting the first and second shipping point and then coming back to the warehouse (from the environment inspection, the route for this truck is described as [0, 1, 2, 0]); the second truck (green line) instead, is traveling [0, 2, 3, 4, 5, 0] and the third truck (red line) is covering [0, 5, 0]. These routes are based on the remaining capacity of each truck, on the number of initial trucks and the demand of each customer. Since the capacity of each truck is equal to 1000 kg and the aggregate demand is greater than the overall capacity of the two trucks, a new vehicle is added to the environment in order to complete the task and serve the remaining packages to load from the shipping point number 5.

The optimization of this first scenario focuses its attention on the attribution of the rewards in this way:

$$Total\ Rewards = Completion\ Rewards - \lambda\ \cdot\ Distance\ Penalty$$

The two components of the total rewards are conflictive, as they may create some trade-offs based on the values assigned to the weighting parameter λ. The trade-offs arise insofar as the realization of one of the objectives (maximizing the rewards and minimizing the penalties) may affect the fulfillment of the other. To avoid these compromises and reach optimization, DMs need to find an equilibrium point at which they can appoint the value of λ. The objective function is then:

$$R_{total} = \begin{cases} C - \lambda \cdot D_{total}\ if\ \sum_{i=1}^{n} f_i = n \\ -\lambda \cdot D_{total}\ otherwise \end{cases}$$

where C is the completion reward, D_{total} is the total traveled distance, n the number of customers and f_i a vector of dimension n (if $f_i = 1$ the customer has been served, otherwise if $f_i = 0$). If $f_i = n$, then all the customers have been visited.

In this first case, the model's hyperparameters have not been implemented and, as a consequence, they have not been tuned. To train the agent in this scenario, the algorithm A2C has been chosen: first, the environment must be wrapped with a monitor function able to supervise the results obtained. The model is then created through the policy model called 'MlpPolicy' and the agent is learning with the function `learn()`. Some episodes can be executed to test the correct performance of the algorithm and to understand potential improvements (especially to the rewards arrangement).

For debug reasons, a script like the one here below can be printed at the end of each episode. The "*rollout*" phase includes all the actions taken in the environment by the model and here statistics such as the mean length (5.43) and the mean reward (47.1) of each episode can be collected. Below the item "*time*", valuable information concerning the training time is displayed. During the "*training*" phase (where the policy is boosted), the model gathers data about the entropy loss (given its low value, this episode counts on a high reliability of the agent's actions [53]). It also specifies the explained variance (the value is not well-explaining the rewards), the learning rate, the number of updates, the policy loss and the value loss. These last two indicators should be constantly monitored throughout the execution of the other episodes: in fact, they should keep decreasing episode after episode so that the difference between the actual policy (value) and the expected policy (value) is minimized.

```
-----------------------------------
| rollout/                |         |
|    ep_len_mean          | 5.43    |
|    ep_rew_mean          | 47.1    |
| time/                   |         |
|    fps                  | 294     |
|    iterations           | 2000    |
|    time_elapsed         | 33      |
|    total_timesteps      | 10000   |
| train/                  |         |
|    entropy_loss         | -0.666  |
|    explained_variance   | -5.72e-05 |
|    learning_rate        | 0.0007  |
|    n_updates            | 1999    |
|    policy_loss          | -0.901  |
|    value_loss           | 1.92    |
-----------------------------------
```

Fig. 39: episode's statistics.

To test the trained model on a complete episode, it is necessary to reset the environment and the needed variables. Afterwards, the model is used to predict the best action and once the episode is terminated, the new vehicles' routes are obtained. The first vehicle is completing the route [0, 1, 2, 0], while the second vehicle is visiting the customers 2, 5 and 4 leaving customer 3 unserved and not coming back to the warehouse. This is happening because the agent is not successfully learning how to execute the actions in the environment. The causes can be related to the fact that no hyperparameter tuning has been already completed, to the low number of iterations (10) and to the structure of the environment (especially the development of the '`calculate_rewards()`' function).

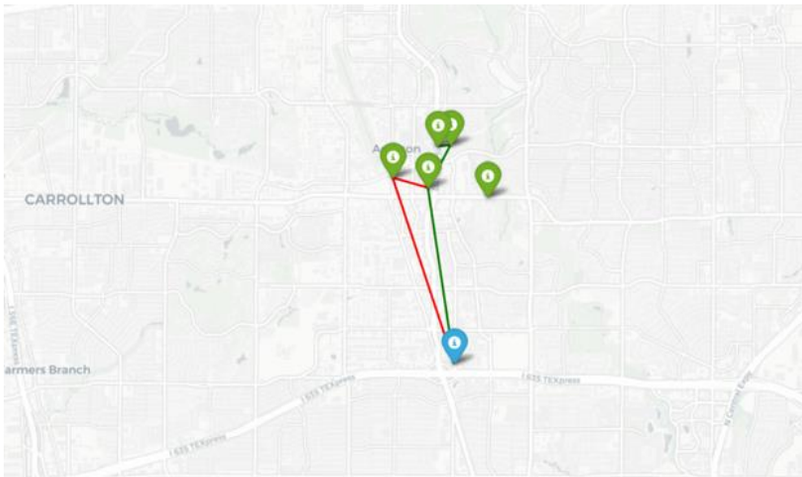


Fig. 40: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

In the second scenario, a shipping point can be served only by one truck, time window constraint is added, the '`calculate_rewards()`' function is revisited and the study of the hyperparameters is improved.

After investigating an episode after the adjustment of the VRP environment, the trucks can correctly visit all the shipping points. The vehicles are also considering the load of the distinct customers and their maximum capacity (1000kg), respecting in this way the enforced initial constraints.

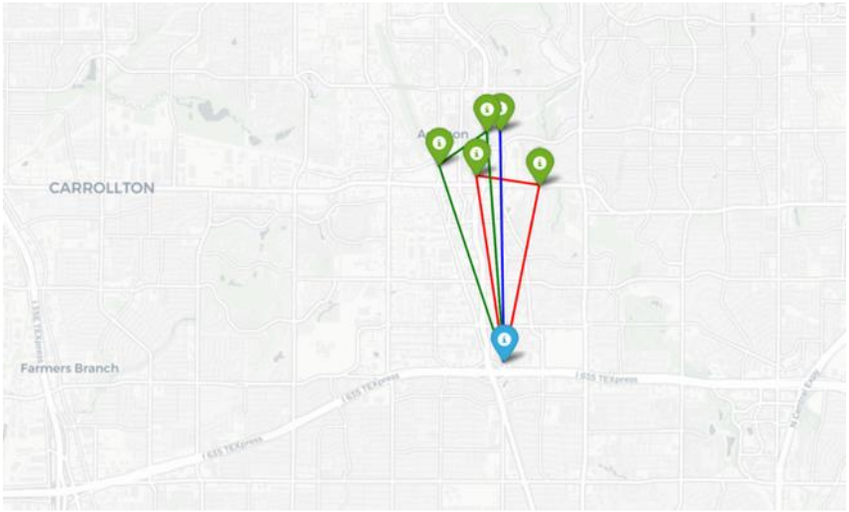


Fig. 41: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

To do so, an additional function called '`next_location()`', has been developed in the environment object. This function finds the potential customers to be visited by a specific truck, among the ones that still have to be visited and that do not exceed the remaining capacity of the truck. In addition, the '`calculate_reward()`' function has been revisited: the agent will receive a penalty whenever the time window constraint is not being observed and a reward for every new vehicle added to complete the task.

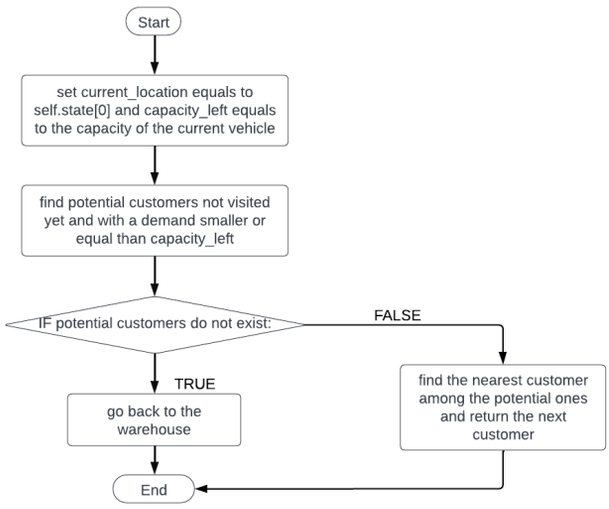


Fig. 42: `next_location()` flowchart.

Here, the rewards per step of the investigated episode before the training phase, are displayed:

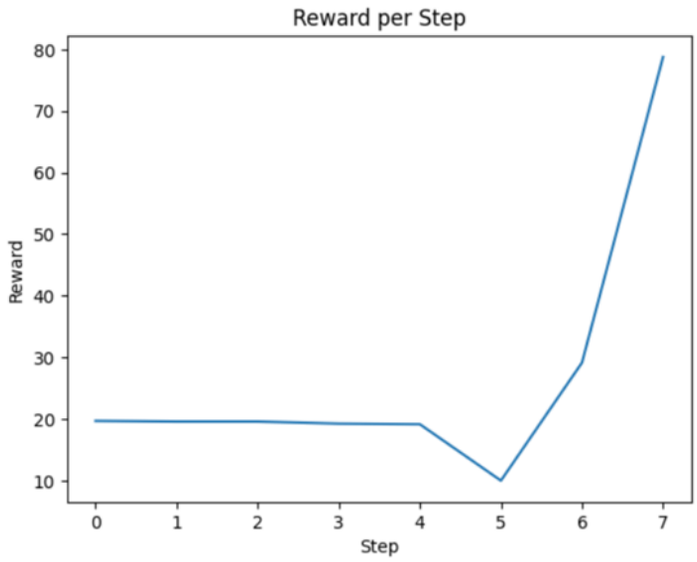


Fig. 43: episode rewards of the environment before training.

From here, the new objective function is:

$$R_{total} = C - (w_d \cdot D_{total} - w_t \cdot \sum_{k \epsilon K} \sum_{l \epsilon R_k} P(k,l)) + w_n \cdot K$$

Where w are the respective weights, $P(k,l)$ is the penalty of vehicle k to point l. The latter is assigned if the vehicle arrives before or after the time windows.

The chosen model to train the agent is still A2C, but now with the implementation of the hyperparameters. Several tests have been run, but the chosen hyperparameters for this study allow to reach a higher explained variance level, that now reaches 0.947. From the image here below, it can be noticed that in this episode, the rewards are well-explained even if the loss value is elevated. The policy seems to be reasonably substantial, given the policy loss which amounts to 9.85e-06.

```
-------------------------------------
| rollout/                |           |
|    ep_len_mean          | 7         |
|    ep_rew_mean          | 205       |
| time/                   |           |
|    fps                  | 334       |
|    iterations           | 1000      |
|    time_elapsed         | 29        |
|    total_timesteps      | 10000     |
| train/                  |           |
|    entropy_loss         | -0.000283 |
|    explained_variance   | 0.947     |
|    learning_rate        | 0.01      |
|    n_updates            | 999       |
|    policy_loss          | 9.85e-06  |
|    value_loss           | 97.1      |
-------------------------------------
```

Fig. 44: episode's statistics.

The hyperparameters chosen are documented in the following table:

HYPERPARAMETERS	VALUE
Policy	"MlpPolicy"
Learning rate	0.01
Entropy coefficient	0.03
Number of steps	2048
Gamma	0.99
Time steps	20000

Table 7: hyperparameters of the A2C model.

The agent is indeed learning with an average reward per episode of about 205. Moreover, it is efficiently managing the new logic and the new rewards/penalty balance.

Now, the study wants to investigate the usage of the PPO model. The environment implemented is the same used for the second scenario, and so is the objective function. From the below image, the model's statistics can be properly analyzed.

```
-----------------------------------------
| time/                   |             |
|    fps                  | 352         |
|    iterations           | 17          |
|    time_elapsed         | 144         |
|    total_timesteps      | 51000       |
| train/                  |             |
|    approx_kl            | 0.010701191 |
|    clip_fraction        | 0.216       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.73       |
|    explained_variance   | 0.942       |
|    learning_rate        | 0.01        |
|    loss                 | 42.5        |
|    n_updates            | 160         |
|    policy_gradient_loss | 0.0105      |
|    value_loss           | 85.7        |
-----------------------------------------
```

Fig. 45: episode's statistics.

As for the other two cases, the first section is dedicated to the "*time*" metrics, including the fps, the number of iterations and the number of total timesteps. In the second part of the statistics, "*training*" metrics can be found: the divergence Kullback-Leibler [54] refers to the relative entropy, pointing to the difference between the old and the new policy. In this case, the value (0.010701191) is low, but when dealing with PPO, it's a metric that should always be monitored: too high values indicate the agent's learning vulnerability [55]. This algorithm employs a loss function, which contains a "*clip*" term that aims to steady the learning of the agent. Here, the two metrics concerning the clipping method are '`clip_fraction`' and '`clip_range`': the first one tells how much the clip range is used (21.6% of the actions has been clipped), while the latter explains the range of the clipping to limit policy updates (± 0.2). Here, the low entropy loss (-1.73) points out that the agent is not appropriately exploring the policy; on the other hand, the rewards are well-explained (as the explained variance is 0.942). The policy loss explains how and how much the policy is transforming based on the gradient and it should be smaller than 1. Instead, the value loss demonstrates the ability of the model to predict the outcome of every state [56].

The hyperparameters chosen are documented in the following table:

HYPERPARAMETERS	VALUE
Policy	"MlpPolicy"
Learning rate	0.01
Entropy coefficient	0.05
Number of steps	3000
Gamma	0.99
Time steps	50000

Table 8: hyperparameters of the PPO model.

Generally, the model is successfully training the agent, even if the room for improvement is still wide. Hyperparameters could be tuned and optimized so as to obtain better model's statistics. The average reward per episode is reaching the same value attained by the A2C model (205) and this could be caused by the ease of the environment or by the number of evaluation episodes.

Before introducing the next paragraph, it has been decided to perform once again the model with the best results (A2C) with a bigger set of data. First, the remaining customers connected to the already studied warehouse are considered for the model resolution. As a matter of fact, the total number of shipping points is now 50 instead of 5 and all the trucks leave from the same warehouse as before.

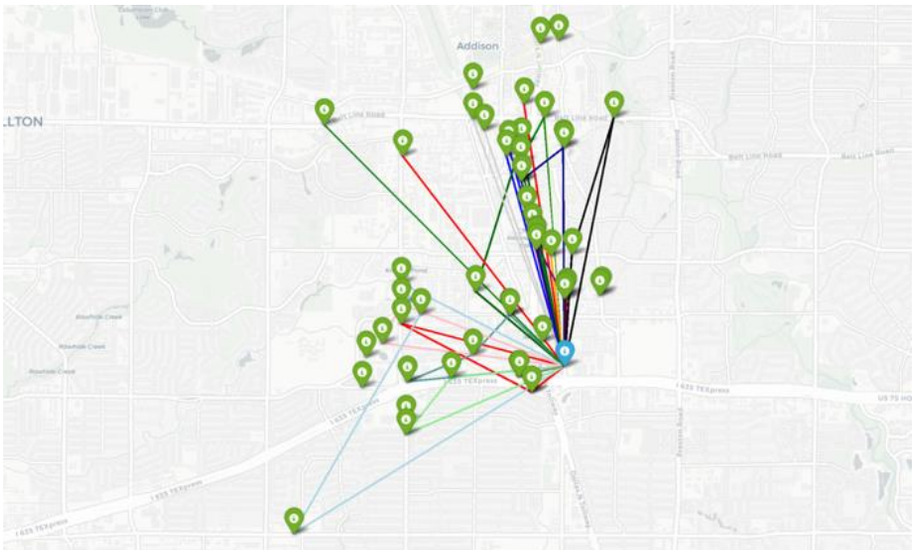


Fig. 46: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

It is noticeable that the paths are not optimal, as some shipping points are ignored by the agent (not visited) and the routes look randomly arranged. From this, it's arising the agent's lack in the generalization of a big set of unseen data. The agent should be able to create clustered paths based on the minimum distances between shipping points and on the trucks' capacity. However, in this case it looks like the only factor taken under consideration is the remaining capacity of the vehicles.

```
-------------------------------------
| rollout/                |          |
|    ep_len_mean          | 69       |
|    ep_rew_mean          | 1.36e+04 |
| time/                   |          |
|    fps                  | 275      |
|    iterations           | 1000     |
|    time_elapsed         | 36       |
|    total_timesteps      | 10000    |
| train/                  |          |
|    entropy_loss         | -2.03e-05 |
|    explained_variance   | 1.19e-07 |
|    learning_rate        | 0.01     |
|    n_updates            | 999      |
|    policy_loss          | 0.00191  |
|    value_loss           | 1.27e+06 |
-------------------------------------
```

Fig. 47: episode's statistics.

These episode's statistics show that the performance of the model is poor and that it doesn't reach the values of the previous implementations. The explained variance is way lower than before, meaning that it can't correctly explain the rewards, which are extremely elevated with respect to the ones encountered before.

Now, the other two warehouses and their respective shipping points are included in the model.

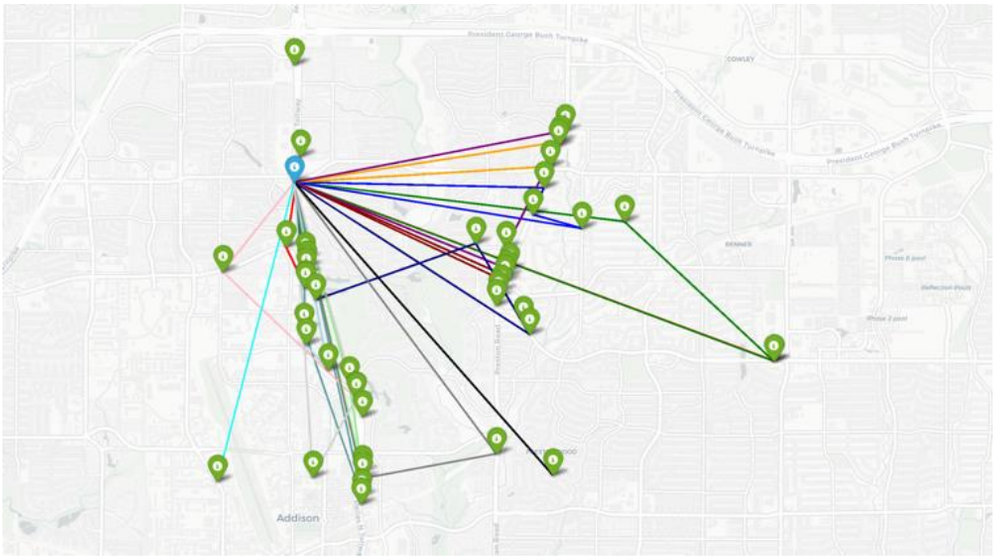


Fig. 48: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Figure 48 shows the optimized routes of the second warehouse of the dataset and its connected shipping points. Here, some of the paths look optimally distributed, while others are going back towards the warehouse and then moving away again, confirming what previously said for figure b. Also in this case, one customer is not visited: this could be a result of the poor study of the hyperparameters of the model.

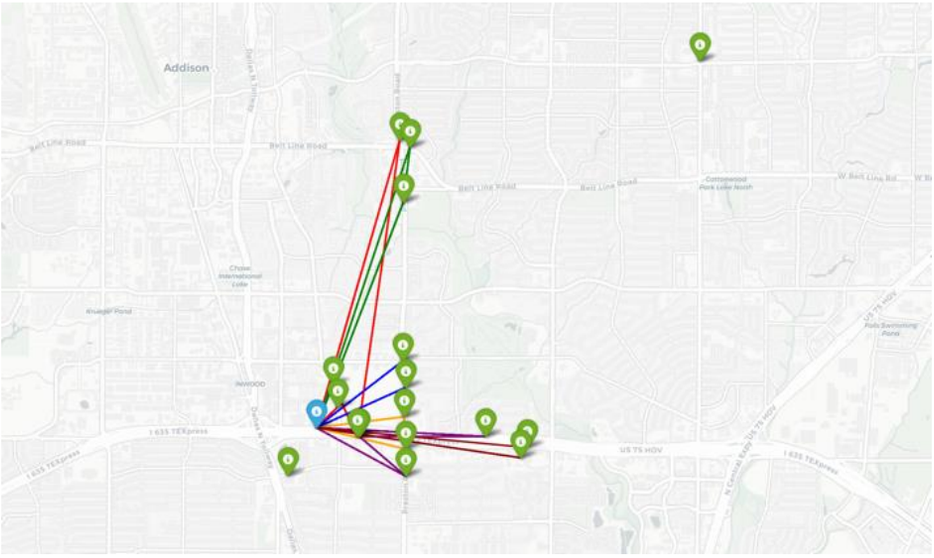


Fig. 49: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

The third warehouse together with its shipping points and the adjusted routes, is represented by figure 49. Here, the number of shipping points is decreased (from around 50 to 16) but what has been discovered so far still applies.

2.2.4 Further refinement.

To improve the models presented in the last paragraph, a deeper hyperparameter study should be carried out. In addition, the reward function could be refined with supplementary penalties and rewards: for example, a penalty could be assigned if the truck remains for too long (more than 40 minutes) in a shipping point to charge the vehicle. Instead, if a truck serves all the allocated shipping points in less time than expected (a threshold is set), then the agent will receive a reward.

The environment should adequately reflect the reality and all the sophistication that comes with it. For this reason, how states and actions are defined in the class object can be reviewed and better solved. Another important aspect to consider, is the relevance of stochastic variables (especially within the scope of PPO): a truck could face a blocked or trafficked road while completing its tasks. This affects its performance and the amount of time needed to serve the shipping points, leading to a failure to comply with the time window constraint. The more the environment is uncertain, the more it becomes complex. When training the agent with the Q-Learning algorithm, for instance, the use of a stochastic policy enhances the exploration of the state [57]. For this reason, it could be interesting to perform a DQN with the Q-Learning algorithm. This might lead to a better generalization of the policies to unseen data, but also to a lack in the policy accuracy [58].

Furthermore, balancing exploration and exploitation (both defined in [59]) is an advancement that should be taken under consideration. [59] also lists some of the possible strategies which help in finding a better balance of these two fundamental dimensions, just as the Epsilon-Greedy strategy and the Decay Epsilon Over Time among others. It's a trade-off that needs constant analysis and refinement to better address the agent towards the correct decision. In this study, to equilibrate the exploration and the exploitation, the rewards and the penalties have been carefully developed trying not to assign more weight to one instead to another and vice versa.

Regularization techniques can be added to avoid the phenomenon of overfitting and to boost generalization [60], especially when dealing with stochastic policies.

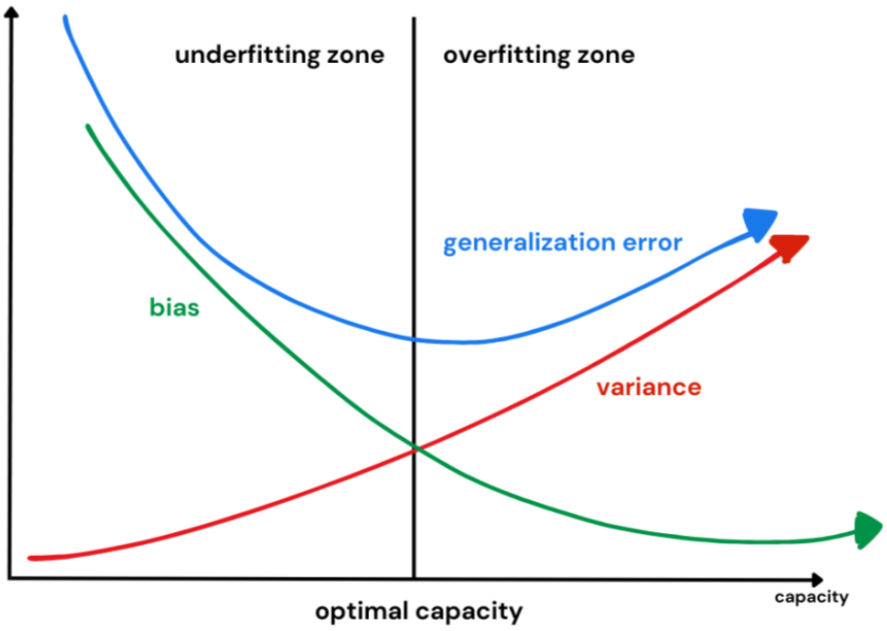


Fig. 50: U. Tewari, Regularization — Understanding L1 and L2 regularization for Deep Learning, *Medium*, November 9th, 2021. [Online]. Available: https://medium.com/analytics-vidhya/regularization-understanding-l1-and-l2-regularization-for-deep-learning-a7b9e4a409bf (recreated image).

Some regularization methods include the entropy regularization, the dropout and the early stopping among others (and they are widely discussed in [61]). By contrast, an empirical study on continuous control has been carried out [62]: the authors wanted to highlight the fact that common regularization techniques are not actually used in the field of RL. Consequently, they presented this study regarding "*multiple policy optimization algorithms on continuous control tasks",* discovering that conventional techniques can genuinely improve policy networks.

Likewise, the architectures of the policy or the value function can be expanded. For example, a recent study [63] developed a CNN (Convolutional Neural Network) for the PPO model, proposing an improving algorithm. The research objective is indeed to enhance an already existing solution with the help of RL and neural networks. Authors compared the results of this solution with the one obtained by using OR-Tools and they have ascertained that the resolving provided by the latter, remains the best one.

Another aspect to include is the constraints analysis: probably, what described in the study of the constraints should be improved and new restrictions added. For example, the maximum number of vehicles for each warehouse could be limited to a certain value based on the total demand. Or else, the total demand for each shipping point could be limited in a similar way: this will enhance the service provided by the company and improve the performance of the model.

2.2.5 Comparison between heuristics and RL.

Now that both implementations have been carried out, an introductory analysis is presented in this paragraph. This comparison does not want to declare which method is the best, but instead it wants to see the main differences that have been encountered throughout the fulfillment of the tasks.

Specifically, the OR-Tools second scenario (without the APIs) and the RL second scenario are put together in the following table and discussed. For the moment, only A2C is taken under consideration, while PPO will be integrated in the third chapter.

Here the table reporting the major dissimilarities between OR-Tools and RL:

OR-TOOLS	RL – A2C
Solves optimization problems.	Interacts with an environment to maximize the rewards.
Bases its theory on operational research.	Bases its theory on reinforcement learning.
Uses heuristics algorithms.	Uses a policy.
Deterministic.	In this case deterministic, but it can also include stochastic elements.
Needs an objective function, constraints and variables.	Needs states, actions and a reward function.
Less prone to overfitting.	More prone to overfitting.
Receives a final solution.	Receives rewards at the end of each step, according to the action taken.
Optimal solution based on the optimization function.	Performance metrics.
Less complex to develop and with initial low implementation costs.	Complex to develop with high implementation costs.

Table 9: differences between OR-Tools and RL-A2C.

The graphical solutions are proposed again to see how the routes change from one implementation to the other. Figure 17 (1.3.2) shows the solution found by OR-Tools, which sees three different routes: the first vehicle serves only one shipping point (3) as the second one (1). The last truck serves three different shipping points (5, 4, 2) while respecting the capacity constraints. All three trajectories leave from the warehouse and come back to it, as expected.

On the other hand, figure 40 (2.2.3) displays another combination of served customers. Here, the first vehicle serves two shipping points (3, 1), the second vehicle serves other two shipping points (2, 4) while the last truck only visits one shipping point (5). They all respect the capacity constraints as well; they start and finish their path from node 0.

The total amount of kilometers traveled with OR-Tools is around 18km, while with RL – A2C is around 20km. This means that the latter is taking a longer path than the first one. In the next chapter, an economic analysis will indicate which system to choose and when.

2.3 Conclusion.

In this chapter, two VRP environments have been created suited for the resolution of the task with RL models. Understandably, the second environment improved the final solution with respect to the first scenario environment. On the other hand, adding more data to the model decreased the efficiency of the solution, which needed further refinement.

Throughout the course of the several sections, some implementation ideas have also been proposed. These want to be a starting point for future research and a way to criticize the work done so far. Breaking down the improvements that could be made, allowed to realize the mistakes committed and how to resolve them theoretically.

A preparatory comparison between OR-Tools and RL has been made in the previous paragraph, while Chapter Three will focus on the RL proposal. Here, a deep economic analysis is reported and presented to a fictitious company, interested in acquiring new methods for the optimization of its processes.

Chapter 3. Analysis of the solutions.

3.1 Introduction.

In this chapter, the RL solution is analyzed and compared to the heuristic methods. In the first place, the proposal is studied through TensorBoard [64], which is a powerful visualization tool able to register the metrics of the models. The final purpose is to show complete graphs that can explain the pure performance in this data context.

Going forward, the economic analysis is carried on. Here, the difficulties that can emerge from the adoption of the method are discussed with the reasons that could convince a company to move towards this solution. Moreover, a SWOT together with a costs and benefits analysis are performed: this will allow a critical debate on the resolution flaws and lacks.

3.2 Metrics and analysis of the models.

With TensorBoard, the following graphs are created to understand the trend of the already studied metrics along the various episodes. To better explore them, it has been asked to ChatGPT to give insights about the progress of the graphs' curves which have been deeply investigated through the available documentation [65].

At the outset, the A2C model concerning one warehouse and only 5 shipping points, is analyzed.

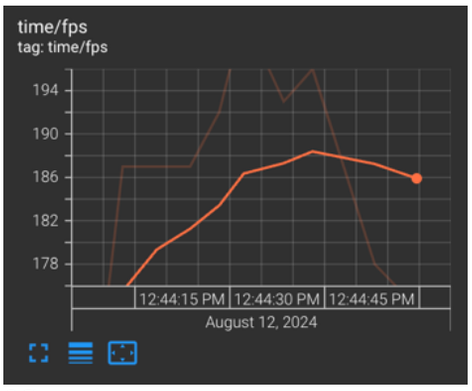


Fig. 51: one warehouse, 5 customers A2C metrics.

FPS stands for Frames Per Second and in this case, it represents the time taken by the agent to fulfill the tasks. The trend of the curve is increasing in the beginning, but it starts decreasing at around 187 frames per second. It looks like the agent is rapidly learning, ensuring a certain balance and stability.

Moving forward, the entropy loss reaches deterministic values indicating that the agent is certain about the decisions taken. The explained variance – as already mentioned before – shows a positive and increasing trend throughout the episodes, meaning that agent's rewards are well explained. The learning rate instead, is stable and constant as the given value is equal for all episodes. The fourth graph displays the curve of the policy loss, which results to be unstable but that is common to notice in models like this. The first decreasing part of the trend demonstrates that the agent is correctly starting to learn the policy; the increasing section instead, shows that the model is going through a difficult phase of the learning journey. Fortunately, it is possible to notice that the policy loss value starts to decrease again once it reaches a peak (around 0.0022): this signifies that the agent found a way to overcome the difficulties encountered before and is now able to enhance the performance of the model.

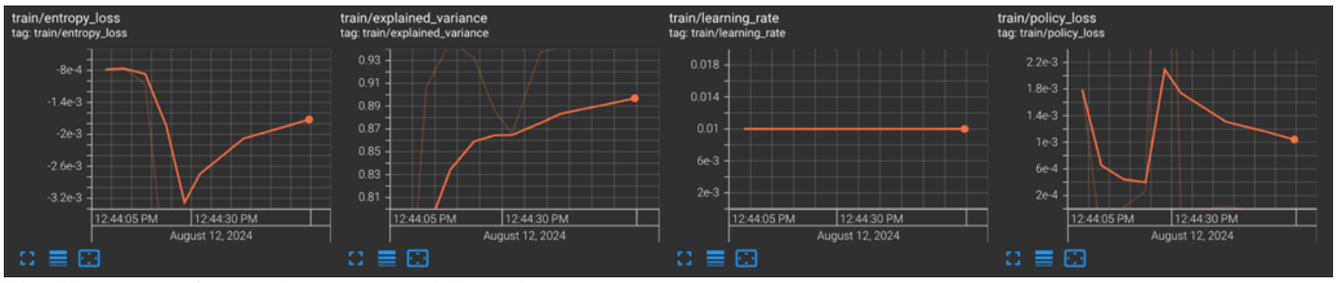


Fig. 52: one warehouse, 5 customers A2C metrics.

Similarly to the policy loss trend, the decreasing value loss trend points out that the agent is correctly estimating the error between the value function and the estimation [65].

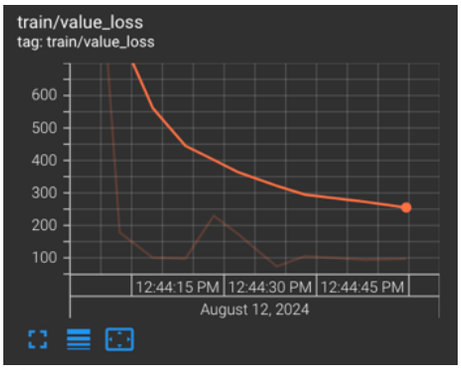


Fig. 53: one warehouse, 5 customers A2C metrics.

The presence of the same reward value in every episode is not a positive factor. As a matter of fact, it looks like the agent is not prompted to learn new actions and to distinguish between those actions leading to a penalty or a reward. In the future, there will be no actual enhancement of the model.

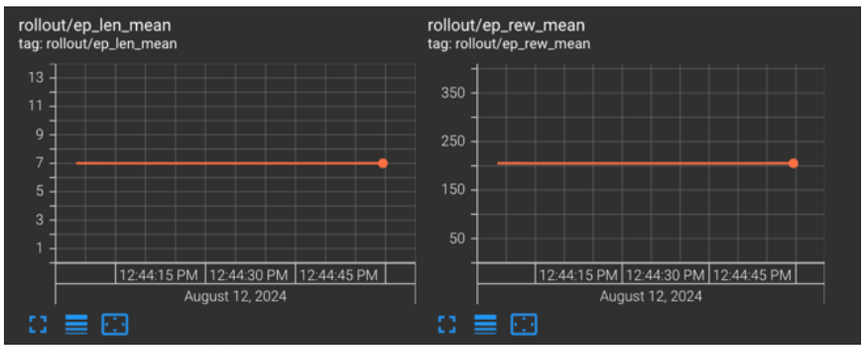


Fig. 54: one warehouse, 5 customers A2C metrics.

More or less, the same trends are also encountered when adding all the remaining shipping points – related to the warehouse – to the model.

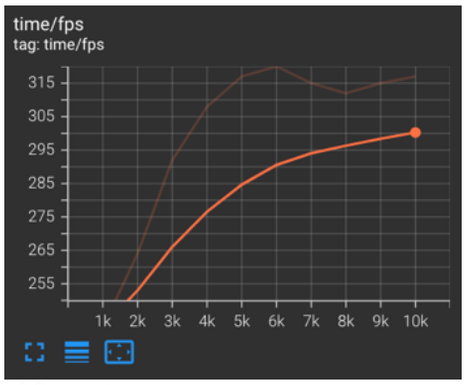


Fig. 55: one warehouse, all customers A2C metrics.

Here, the explained variance floats between small positive values: this indicates that the rewards are not fully explained by the model. Now, the policy loss only follows a decreasing trend that seems to show that the model is learning the policy without any problem.

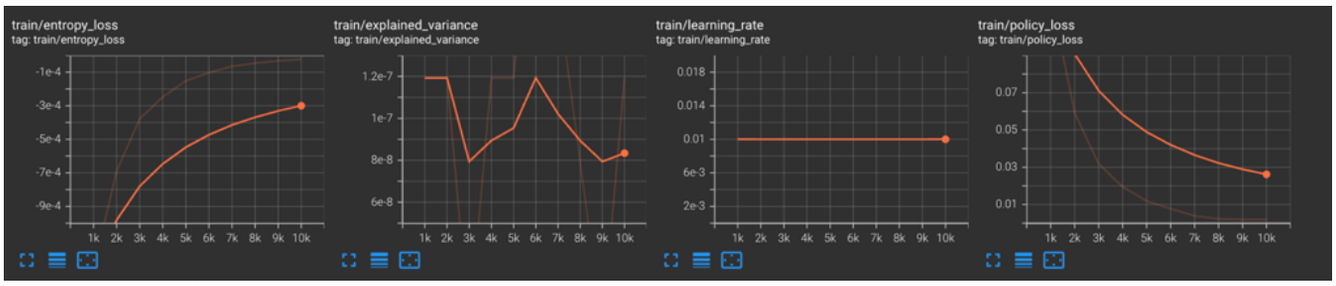


Fig. 56: one warehouse, all customers A2C metrics.

Fig. 57: one warehouse, all customers A2C metrics.

Once again, the same reward value is found for all the episodes. As already stated for the previous graphs, this is not an indicator of an optimized A2C model. To tell the truth, this model still has a lot of work to be performed on, starting from the environment and the rewards assignment to the hyperparameters study.

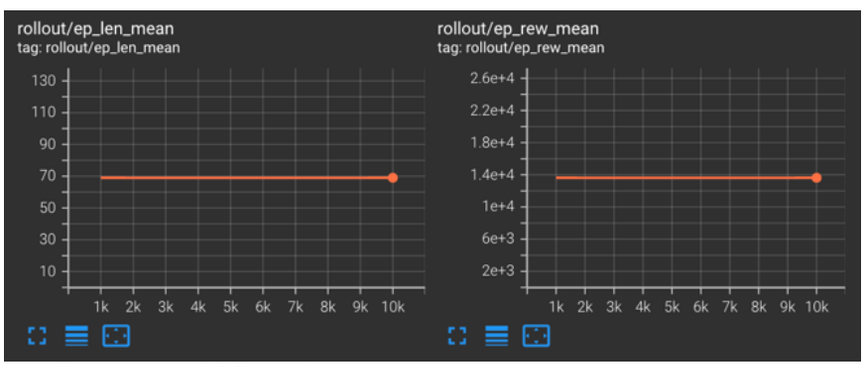


Fig. 58: one warehouse, all customers A2C metrics.

The metrics showed so far, leave big room for improvement and further refinement. Especially the development of the environment in which the agent operates, should be carefully analyzed to understand which logics are to be modified or added. As a matter of fact, the environment class is the most important element of the entire code and for this reason, it needs deeper attention.

3.3 Economic analysis.

In this section of the last chapter, the economic analysis is split into three micro areas: SWOT analysis, costs and benefits analysis and scenario planning about future possibilities.

Before digging in, several companies have already decided to incorporate RL and its benefits in their day-to-day operations. This method is becoming more and more appreciated, especially in the Supply Chain Management (SCM) sector [66], the main topic of this thesis. For instance, Amazon [67] developed a new RL platform called DeepRacer, able to ease the optimization of the autonomous vehicles' routes. In addition, Amazon is starting to deliver packages through independent drones created thanks to the RL technology. Therefore, RL is not only exploited to optimize the paths of shipping vehicles to minimize costs and maximize the efficiency; instead, it finds a wide employment in the creation of the mechanism – of drones, in this case – that will eventually bring the desired outcomes.

So far, it has only been discussed about the delivery of material goods, but it also exists for the transportation of people. Usually, it has always been assigned to Taxis; since a few years, with the advent of Uber and other Transportation Network Companies (TNC), this type of service has been shifted to mobile applications. This transition is not only facilitating the communication between the driver and the passenger, but also decreasing general costs and CO2 emissions. The main objective of TNC though, is to optimize the scheduling of vacant vehicles to successfully match them with demanding passengers, avoiding all types of delay and maximizing drivers' revenue. These vehicles are called "*cruising vehicles*" and [68] reserves a special attention towards them: as a matter of fact, the authors developed a scheduling solution based on a DQN algorithm. This study follows three key points:

- DQN must learn the urban area of interest.
- The optimization aims at maximizing the long-term revenue of the drivers.
- The model is trained on big real-time data and historical data.

As a result, the searching time for a ride is shortened and long-term revenues are increased, as shown by the performance evaluation carried out by [68].

Going back to the focus point of this work, it has been wanted to address the proposed solution to a fictitious company (named A, for simplicity) keen on investing in innovative methods. Company A is a giant in the delivery market, especially in the European one: unfortunately, the high costs it had to face during the last two years, made company A lose several clients. Those costs include the rising price of the gasoline and the trucks' maintenance: in fact, some of the resources owned by company A result old and in need of constant mechanic checks. After a long analysis of the situation, the Executive Board noticed that the significant costs could have been avoided if the company had an optimized and scheduled plan for the deliveries. Until now, company A has never optimized the management of the vehicles sent from the warehouse to collect the customers' packages. This could have helped in saving the vehicles' fuel and in the preservation of the trucks (which now must be changed). Now, company A wants to improve its decision-making model starting from optimizing the paths of the vehicles based on the traveled distance, the capacity, the fuel consumption and the number of available trucks. The board decided to set an approximate period of one year and a half to fulfill the resolution of their internal crisis. Some of the analysts of the group found attractive solutions concerning both heuristic and RL implementations. Thanks to the next paragraphs, company

A will better understand the pros, the cons, the effective costs and benefits of this new way of operating.

3.3.1 SWOT analysis.

SWOT analysis is one of the most powerful tools in the context of strategic planning. It highlights the relevant internal and external factors that define the object of the study. In addition, it has been tried to guess the opinion of the different stakeholders involved in the process. Here, two SWOT analyses are provided: one concerning the heuristic solution and one concerning the reinforcement learning proposal.

Fig. 59: heuristic solution SWOT.

First, company A wants to know more about the heuristic solution. At the first glance, this implementation seems to be quite interesting: in point of fact, from one side it doesn't require high entry costs and from the other, it furnishes precise and optimal solutions. On the contrary, it highlighted the fact that it doesn't work well with real-world scenarios – which is what company A needs now.

Fig. 60: RL solution SWOT.

RL looks more promising from the obtained SWOT. Starting from the strengths, RL models work well with large amounts of real-world data and can achieve high levels of accuracy. Contrariwise, to reach the desired objectives, a substantive volume of resources is required, including:

- Entry costs.
- Training investments.
- Time to successfully optimize the proposed model, suited for company A exigencies.

By contrast, this system could lead to suboptimal solutions. This is the case faced so far: in fact, the solution proposed using RL is still not providing optimal paths.

From the solutions obtained until now, company A can only understand the power of RL in the long term. For that matter, RL is working – approximately – well with a small quantity of data while it needs a finer tuning for when dealing with a bigger dataset.

In response of these considerations, it would be interesting to propose some methods to surmount the difficulties encountered and to exploit the opportunities. A well-defined scheme is below proposed:

- Heuristic solution.
    - How to mitigate and handle threats: since the threats are related to the non-availability of real time data, company A could hire a new team responsible for analyzing and predicting traffic conditions. The geocoding scope – explained in [69] – issue arises because the tools implemented don't offer an integrated geocoding system. As already done in Chapter One, APIs (OpenStreetMap) can be used to overcome this threat. Moreover, this method can be harnessed in the testing phase with smaller sets of data.
    - How to leverage opportunities: Mitigating a threat results in an opportunity, which is the possibility to implement APIs in the model for the geographic positions. The availability of information regarding the routes, helps in the real time data threat. The new team could exploit this opportunity by gathering as much as possible data and creating predicting models.
- RL solution.
    - How to mitigate and handle threats: companies, – especially the older ones – might be skeptic about investing in new methods such as RL. For this reason, the proposal should be already optimized and producing promising results. It should come together

with a detailed economic analysis and real cases of businesses in which this solution succeeded. The possibility of generating non optimal solutions is the one that could scare the most the team and the investors: a backup plan (heuristics, in this case) is always important to have, together with constant monitoring (3.3.2 and 3.3.3 widely describe this threat).

- How to leverage opportunities: apart from the skepticism of some companies, as many are eager to invest in innovative technologies because of their novel needs. This should be exploited by operating on a specific target – which can include startups or new businesses – and developing ad-hoc solutions with fresh ideas. Another important aspect is the wide presence of models' documentation: this can be used to overcome the threat of not reaching optimal solutions. Training on the new method is a powerful instrument both for mitigating threats and leveraging opportunities.

In the next paragraphs, the focus will be on the RL method, which is the one chosen by company A for now.

3.3.2 Costs-Benefits Analysis.

In this paragraph, a Cost-Benefit Analysis (CBA) for the RL solution is carried out. Thanks to this investigation, company A will know if the costs of the proposed solution are higher than the actual benefits it is going to bring. When this is the case, it means that the business should invest in the RL proposal. This CBA will also include the evaluation of opportunity costs, which give a greater level of accuracy in the decision-making process.

During the first phase, as also suggested by [70], the project scope is defined. The goal of this CBA is well-defined in 3.3 and it can be summarized as "*minimizing transportation costs through the optimization of the vehicle routes*". [70] proposes some key points to take into consideration to better formulate the scope of the project:

- Timeline: as already said, the Executive Board of company A, set a deadline of one year and a half starting from now to successfully implement the final model. The proposed timeline is represented here below and it integrates five important steps to go live with the new RL model.

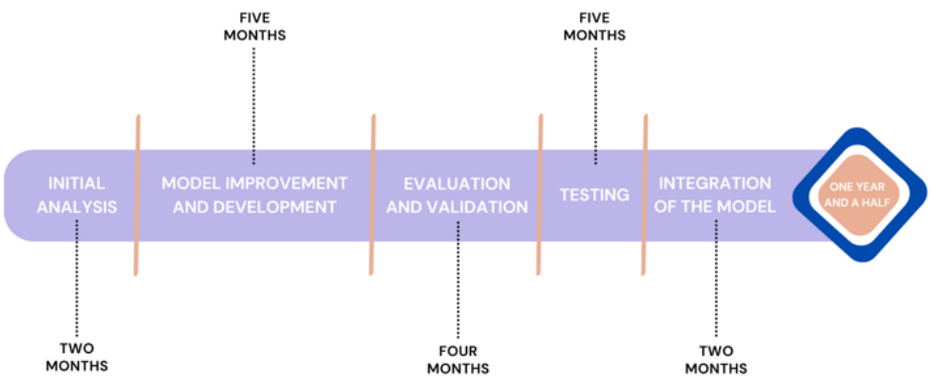


Fig. 61: proposed timeline to implement the RL solution.

The initial analysis is taking place right now, with the SWOT, the CBA and the scenario planning and it should last two months at maximum. Moving forward, there should be an important phase of model improvement and development: as a matter of fact, the model right now presents some discrepancies and should be arranged and customized according to company A's exigencies. The evaluation and validation step together with the testing phase are the most fragile procedures because they are the ones that tell if the project will take the

right turn. Finally, the integration of the model in the decision-making process is the last phase of this timeline. Here, the project goes live and starts operating in real-world scenarios.

- Resources: apart from substituting the old trucks with a new electric fleet, the resources needed include highly trained employees (a team of 10 people), especially engineers and developers. External resources could be appropriate during the timeline. To do so, a substantial capital is needed to cope with the technical costs, the wages, the cost of the RL proposal and the truck investment.
- Constraints: the Executive Board decided that if after the first two phases of the timeline, the team can't propose efficient solutions, then the project is said to be closed and it would be needed to start again. Moreover, the available sum for this project to go live (except for the investment of the new fleet) is 1,200,000 euros, which should be split among the resources.
- Evaluation techniques: some metrics that are going to be used in this CBA comprehend the Return On Investment (ROI), the Benefit-Cost Ratio and the Net Present Value (NPV). Another important indicator is the difference between the benefits and the costs.

Company A is also understanding which stakeholders to involve during this initial analysis, as they will be influenced by the new project. From a detailed consideration, it emerged that the IT and the logistic departments are the ones that could experience a stronger impact, together with the legal affairs team. Their involvement will certainly give the possibility to present a more accurate and precise analysis of the actual costs.

Now, costs and benefits are determined considering an amount of time of 18 months (one year and a half). The predicted costs can be summarized as follow:

- Price of the proposed RL solution → 15,000 euros.
- Implementation and training costs → 20,000 euros considering a team of 10 people.
- Maintenance costs → 25,000 euros.
- Human capital → 700,000 euros.
- Electricity and utilities costs → 15,000 euros.
- Opportunity costs → 10,000 euros.

The AI of ChatGPT proposes some additional costs that should be included:

- Migration costs → 10,000 euros.
- Testing costs → 20,000 euros.
- Data protection costs → 30,000 euros.

The total amount of predicted costs for this project is 845,000 euros (the budget of 1,200,000 euros leaves space for other investments).

On the other hand, the predicted benefits are:

- Enhancement of employees' hard skills → 85,000 euros.
- Ability to use RL also for other types of problem → 200,000 euros.
- Competitive advantage → 150,000 euros.
- Acquisition of new clients and recovery of the ones lost → 200,000 euros.
- Increased revenues → 300,000 euros.

Once again, ChatGPT proposed other benefits for this CBA:

- Reduction of operational costs → 150,000 euros.
- Reduction of CO2 emissions → 20,000 euros.
- Enhancement of customer satisfaction → 75,000 euros.

The total amount of predicted benefit for this project is 1,180,000 euros. The net benefit instead, is obtained from the difference between the total costs and the total benefits and it sums up to 335,000 euros.

From this information, evaluation techniques can be performed:

- The first evaluation technique is ROI, which measures the effective return on the discussed investment. In this case, for every euro spent, a return of 39.64% will be generated.

$$ROI = \frac{Net\ Benefit}{Total\ Costs} \cdot 100 = \frac{335{,}000}{845{,}000} \cdot 100 = 39.64\%$$

- The Benefit-Cost Ratio compares the predicted benefits and costs to understand the advantage of the proposed project. The obtained value highlights the profitability of the investment.

$$Benefit - Cost\ Ratio = \frac{Total\ Benefits}{Total\ Costs} = \frac{1{,}180{,}000}{845{,}000} = 1.396$$

- The NPV [71] is another important CBA indicator. It refers to the availability of current and future cash. Positive results – as the one achieved now – demonstrate the efficiency of the project and therefore, the board should opt for investing.

$$NPV = \sum_{t=0}^{T} \frac{B_t - C_t}{(1+r)^t} = \frac{1{,}180{,}000 - 845{,}000}{(1+0.08)^{1.5}} = 298{,}475.83$$

  Where:
  - T is Time.
  - t indicates the number of years to dedicate to the project (one year and a half).
  - r refers to the discount rate (8%).

At this point, it would be reasonable to undertake a sensitivity analysis. This type of study wants to provide insights about how minor changes in what indicated so far, could influence the estimation results. Specific costs and benefits are crucial in this step, as the evaluation techniques will be recalculated given the alteration of the following variables:

- Maintenance costs.
- Human capital.
- Testing costs.
- Competitive advantage.
- Increased revenues.

Each of these key points will undergo both a positive and negative alteration of 10% and 20%:

Cost/Benefit	**-20%**	**-10%**	**+/-0%**	**+10%**	**+20%**
Maintenance costs.	20,000	22,550	25,000	27,500	30,000
Human capital.	560,000	630,000	700,000	770,000	840,000
Testing costs.	16,000	18,000	20,000	22,000	24,000
Competitive advantage.	120,000	135,000	150,000	165,000	180,000
Increased revenues.	240,000	270,000	300,000	330,000	360,000

Table 10: alterations of costs and benefits for sensitivity analysis.

Now, the evaluation techniques are performed again, once for each alteration of every variable.

Maintenance costs	ROI	B-CR	NPV	BENEFITS	COSTS
-20%	40%	1.40	302,930.70	1,180,000.00	840,000.00
-10%	40%	1.40	300,658.72	1,180,000.00	842,550.00
0%	40%	1.40	298,475.83	1,180,000.00	845,000.00
10%	39%	1.39	296,248.40	1,180,000.00	847,500.00
20%	39%	1.39	294,020.97	1,180,000.00	850,000.00

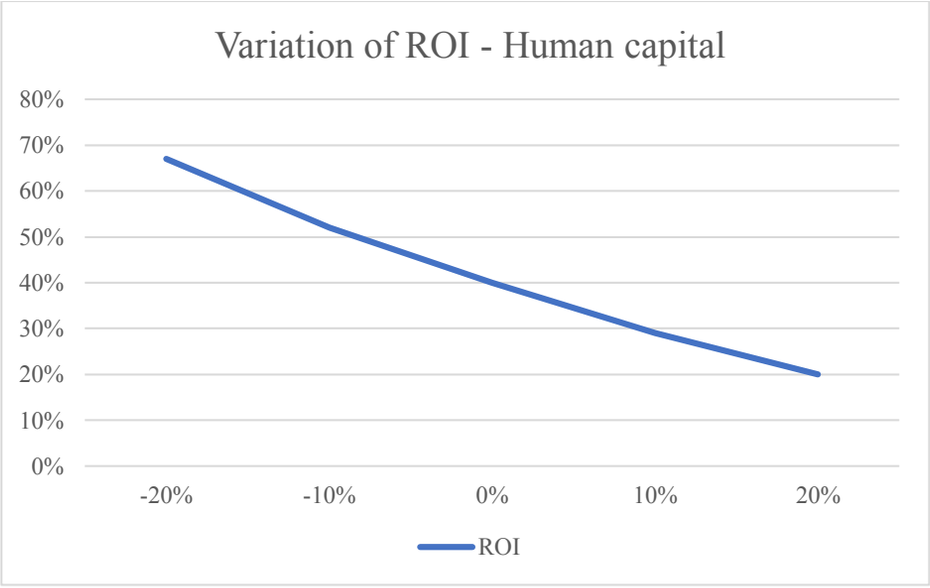
Table 11: evaluation techniques alteration for maintenance costs.

As it is possible to notice, the variation of maintenance costs doesn't negatively or positively affect the metrics obtained with no alterations. Of course, a reduction in this type of costs could expand the possibility to get higher net benefits or to invest more money in the project.
Looking at the data originated from the variations on the investment for the needed workforce, it is noticed that a reduction of 20% increased the ROI to 67%. This could make investors think about how many people they should hire in the team.

Human capital	ROI	B-CR	NPV	BENEFITS	COSTS
-20%	67%	1.67	423,212.00	1,180,000.00	705,000.00
-10%	52%	1.52	360,843.92	1,180,000.00	775,000.00
0%	40%	1.40	298,475.83	1,180,000.00	845,000.00
10%	29%	1.29	236,107.75	1,180,000.00	915,000.00
20%	20%	1.20	173,739.66	1,180,000.00	985,000.00

Table 12: evaluation techniques alteration for human capital.

From the following graph, the variation of ROI is well-displayed and more understandable.



Graph 1: variation of ROI – Human capital.

Similarly to Table 11, the positive and negative alterations on testing costs doesn't bring any change to the attention.

Testing costs	ROI	B-CR	NPV	BENEFITS	COSTS
-20%	40%	1.40	302,039.72	1,180,000.00	841,000.00
-10%	40%	1.40	300,257.78	1,180,000.00	843,000.00
0%	40%	1.40	298,475.83	1,180,000.00	845,000.00
10%	39%	1.39	296,693.89	1,180,000.00	847,000.00
20%	39%	1.39	294,911.94	1,180,000.00	849,000.00

Table 13: evaluation techniques alteration for testing costs.

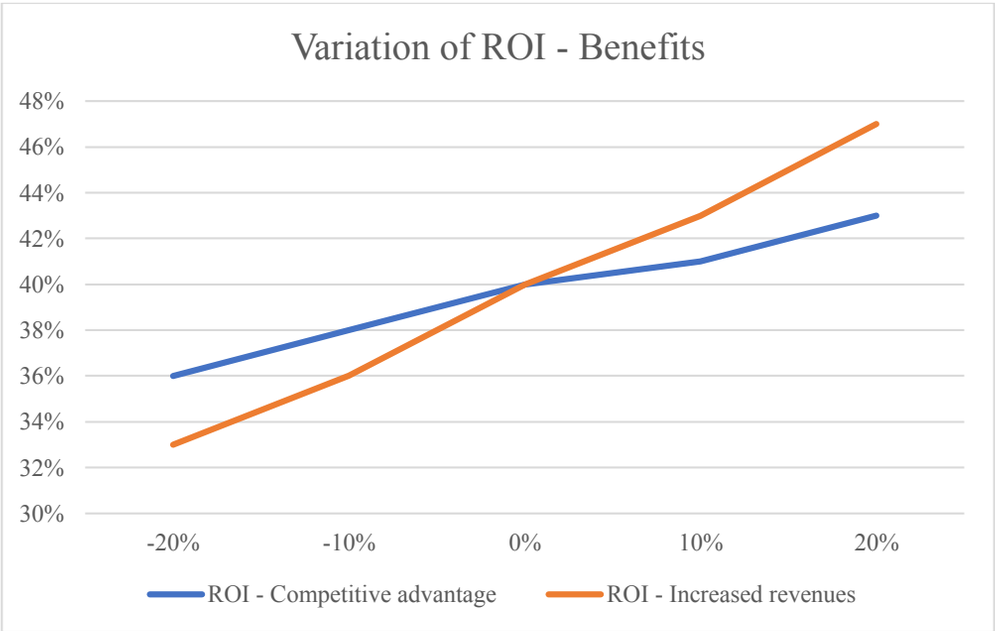
Now, the two benefits (competitive advantage and increased revenues) are analyzed.

Competitive advantage	ROI	B-CR	NPV	BENEFITS	COSTS
-20%	36%	1.36	271,746.65	1,150,000.00	845,000.00
-10%	38%	1.38	285,111.24	1,165,000.00	845,000.00
0%	40%	1.40	298,475.83	1,180,000.00	845,000.00
10%	41%	1.41	311,840.42	1,195,000.00	845,000.00
20%	43%	1.43	325,205.01	1,210,000.00	845,000.00

Table 14: evaluation techniques alteration for competitive advantage.

Increased revenues	ROI	B-CR	NPV	BENEFITS	COSTS
-20%	33%	1.33	245,017.48	1,120,000.00	845,000.00
-10%	36%	1.36	271,746.65	1,150,000.00	845,000.00
0%	40%	1.40	298,475.83	1,180,000.00	845,000.00
10%	43%	1.43	325,205.01	1,210,000.00	845,000.00
20%	47%	1.47	351,934.19	1,240,000.00	845,000.00

Table 15: evaluation techniques alteration for increased revenues.



Graph 2: variation of ROI – Benefits.

If the benefit of increased revenues reaches a value of 330,000 euros (+10% of what predicted), it will be enough to reach a ROI of 47% (instead of 40%). On the other hand, even if this variable suffered a decrease of 20% (240,000 euros), the ROI would still generate a positive outcome equal to 33%.

With the view of completing this CBA, another important study should be carried out: the risk analysis. Examining the potential risks that company A could face if it decided to go on with the RL project, is an essential task and all stakeholders should collaborate. In fact, every relevant team is involved in the research and in the drafting of a strategic plan to avoid or overcome such difficulties.

After a brainstorming phase, the following table resumes the feasible risks associated to this proposal.

Risks	**Predicted probabilities**
The proposed model needs more time to be operative.	20%
The proposed model becomes inefficient for company A's needs.	15%
Data regulation risks.	10%
The actual costs exceed the actual benefits (overestimation of benefits).	25%
Employees need more time to understand the model.	10%

Table 16: possible risks.

As already seen, at the moment the model is not working correctly on a wide set of data and, for this reason, it needs time to be accurately adjusted. This could interfere with the timeline proposed by the Executive Board, leading the connected risk to happen to a probability of 20%. Moreover, it could happen in the future (15% of probability) that the RL solution results to be inefficient for the later needs company A may arise. At the same time, data regulation risks can be widely mitigated thanks to the highly qualified experts in terms of data protection. Another consistent risk (with a probability of 25%) is represented by the fact that the CBA could have overestimated the predicted benefits, leading to the possibility to get a negative NPV; this is related to the chance that employees may need more time to get trained, which can increase training costs and wages.

In order to mitigate the risk, the teams together came up with some strategies:

- A constant monitoring of the current situation should always be carried out (almost every day). If new risks emerge, the Executive Board and stakeholders should be immediately informed and new strategies implemented.
- In the first phase of the timeline, other solutions should be considered so as to get a complete overview of what is available in the market. In the meantime, though, the selected cooperators can already start with their job.
- Any type of personal data involved with the project should be previously anonymized (or other data protection techniques should be used, according to the specific situation).
- Costs should be kept lower as much as possible with respect to the predicted ones.
- Training should start immediately and it should be targeted in line with the hard skills already owned by the employees.

3.3.3 Scenario planning.

In this paragraph, future scenarios are brainstormed and analyzed. Here, a scenario planning template is developed and each of the different cases is evaluated. In this way, it will be easier for the stakeholders to elaborate focused and careful strategies [72].

Before creating the template, it is necessary to identify some factors and driving forces:

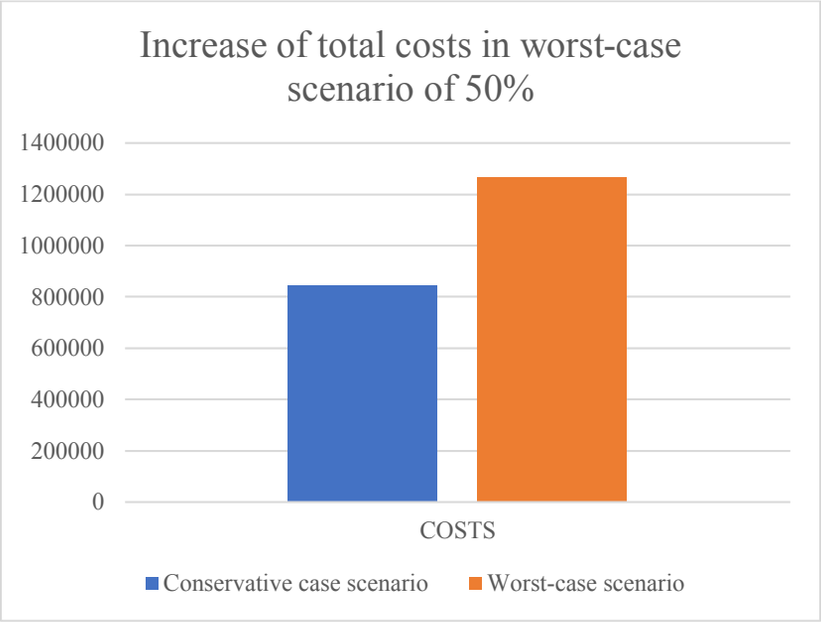
- The time frame, which is projected from 10 to 24 months starting from now.
- Competitors, who could have already implemented a model similar to the proposed one. It is important to understand their results and if it is possible to get better outcomes.
- Possible changes in the demand.
- Potential problems or transitions in the internal workforce.

- The availability of needed data, which should also reach high-quality standards.

ChatGPT suggests developing this analysis by determining three possible scenarios: a best-case scenario, a worst-case scenario and a conservative scenario. By adjusting the several combinations, the different eventualities are specified and analyzed.

The best-case scenario would see the maximization of the difference between benefits and costs (3.3.2). This can only happen if the time spent on the project drastically reduces because of the fast and successful results obtained. In this way, the investment on the workforce can be brought down and this would improve the ROI and the overall perception of the project. Also training costs are reduced, as the team already received the required education about the model; this can result in the enhancement of employees' hard skills (one of the benefits). This scenario would also see the fast implementation of the model, which is highly optimized and tailored for company A's exigencies. The improvement of the solution was fastly brought to the board, which expressed its satisfaction on the chosen resolution to carry the project on. Competitors, on the contrary, are failing with their daily tasks and company A's previous clients decided to come back. Moreover, the needed data is always available and the constant monitoring makes the information premium quality. If this is the real case, the company would improve its performance and the sales would rapidly increase in the short term. RL would also be used for other project ideas and new profitable collaborations could take place.

Conversely, in the worst-case scenario the company would suffer a severe cost increase. This is caused by the fact that the model doesn't work as it should and the team doesn't dispose of high-quality data. Moreover, investors didn't leave the possibility of acquiring other solutions open (as it was previously proposed). This affects the performance of the scheduling optimization, which needs more time to be implemented in the real-world case. Additional time means extra costs, which go up to 40/50% and decrease the ROI to (at least) -7%. Benefits are lowered, especially the competitive advantage and the enhancement of the employees' hard skills. At this stage, employees start feeling demotivated because they are not receiving appropriate training and they are looking for other opportunities in the company competitors. Competitors are taking more and more space in the market, making the presence of company A useless. At this point, looking at the final balance, company A decides to close: the costs incurred for the renewal of the electric fleet have been too high with respect to the annual profits. To avoid this worst-case scenario, employees responsible for monitoring the trend of the project should always analyze and understand from the beginning what is going well and what could take a turn for the worse. For example, if the rest of the team doesn't keep looking for better solutions in the first part, they should constantly bring the point to the table and actively start the research. Instead, if data results are low-quality, they should come up with a successful strategy to fastly retrieve the essential requirement. For instance, they could contact their clients and ask for specific permissions.



Graph 3: increase of total costs in worst-case scenario.

The conservative scenario sees the project going on without any difficulty, but also without any extraordinary result. What has been predicted so far through the CBA results is correct and the project is carried out in the set amount of time. Competition is high, because also other companies reach high levels of optimization: with time, company A can surely exceed their results. As a matter of fact, the team is satisfied with the training received and motivated in taking part in similar projects. Available data complies with company A's demand and potential risks are persistently monitored by whom it may concern.

3.4 Conclusion.

In this last chapter, a deep analysis of the RL proposal has been performed. The key point that came out of it, is that the RL model needs to be improved. In point of fact, it still can't correctly generalize unseen data and this intensely affects its performance. This problem could damage the course of the project of company A, and this is why it should be improved as soon as possible and kept monitored at every change.

On the other hand, if the resolution proves to be efficient and successful, company A could aim at the best-case scenario described in 3.3.3. RL is a powerful instrument that, even if still not widely used by a lot of businesses, promises to completely change the way of working and optimizing complex processes.

Conclusion.

The scope of this thesis was to describe a new solution to solve a VRP, different from the heuristic methods already known. Moreover, the author wanted to develop an economic analysis able to criticize the work done so far and to draw strategic solutions for a fictitious company.

In this last part of the project, it can be said that a RL resolution has been proposed and widely discussed together with a first greedy implementation. Even though the expected successful results have not been reached by the author, both the environment and the model worked and provided interesting outcomes. From these, it was possible to understand that further refinement can be introduced: as a matter of fact, future studies can indeed improve the proposed solution.

Even if several businesses are still insecure about this new method, as many decided to move towards innovation and to optimize their internal processes through the use of different algorithms. The reported examples of Amazon and Uber utilization of RL techniques, supported the grounds of this thesis, which are:

- Providing alternative answers.
- Encouraging businesses to invest in innovation.
- Thinking in a different way.

In addition, criticizing what has been done and studying what instead could have been done, is another important point of – especially – the last chapter. Thanks to the economic analysis carried out through the SWOT, the CBA and the scenario planning, the author could produce an overall picture of how the RL solution can boost the performance of a business. At the same time, if the resources allocation is not optimally adjusted, the investment in such an "*uncertain*" method could be risky for the company.

As a result of this thesis, the author could enhance her analytical thinking and practice her technical skills. In addition, the author deployed her economic studies to create a strategic plan for an imaginary business proposal and reviewed the work done to suggest future progress.

Appendix.

Figures

Fig. 1: F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, M. Yuan, 1st March, 2023. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. [Online]. Available: https://arxiv.org/abs/2303.04147v1 (recreated image).

Fig. 2: V. Vaghela, *Medium*, November 23rd, 2020. Introduction to NP Completeness. [Online]. Available: https://vips3201v.medium.com/introduction-to-np-completeness-d3dfa771d994 (recreated image).

Fig. 3: subtours.

Fig. 4: geo_distance1() flowchart.

Fig. 5: for loop shipping_point_assignments flowchart.

Fig. 6: for loop grouped_shipping_points flowchart.

Fig. 7: for loop group1 flowchart.

Fig. 8: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 9: for loop gx_w1 flowchart.

Fig. 10: add_warehouse() flowchart.

Fig. 11: calculate_distances() flowchart.

Fig. 12: assign_groups() flowchart.

Fig. 13: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 14: create_data_model() flowchart.

Fig. 15: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 16: cost_function() flowchart.

Fig. 17: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 18: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 19: vehicles_function_API() flowchart.

Fig. 20: pickup_function() flowchart.

Fig. 21: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 22: for loop cost flowchart.

Fig. 23: for loop vehicles' schedule flowchart.

Fig. 24: S. Halim, L. Yoanita, Adjusted clustering Clarke-Wright Saving Algorithm for two depots-N vehicles, *Semantic Scholar*, December 1st, 2015. [Online]. Available: https://www.semanticscholar.org/paper/Adjusted-clustering-Clarke-Wright-Saving-Algorithm-Halim-Yoanita/d921cd73380eed72c69ae4b34523d2fb8cfbebad (recreated image).

Fig. 25: Bill, R., Fleck, M., Troya, J. et al. A local and global tour on MOMoT, Softw Syst Model 18, 1017–1046 (2019). [Online]. Available: https://doi.org/10.1007/s10270-017-0644-3 (recreated image).

Fig. 26: F. Simsir, D. Ekmekci, A metaheuristic solution approach to capacitied vehicle routing and network optimization, Sciencedirect, January 23rd, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098618320962

Fig. 27: A. Karthikeyan, Artificial intelligence: machine learning for chemical sciences, Researchgate, September 14th, 2022. [Online]. Available: https://www.researchgate.net/publication/349058264_Machine_Learning_methods_for_solving_Vehicle_Routing_Problems (recreated image).

Fig. 28: Vijay Kanade, What Is the Markov Decision Process? Definition, Working, and Examples, Spiceworks, December 20th, 2022. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-markov-decision-process/ (recreated image).

Fig. 29: The elements of the MDP model for a VRP/VRPTW.

Fig. 30: Q-Learning flowchart from [35].

Fig. 31: The normal and the unknown distribution patterns.

Fig. 32: Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, Martin Takáč, Reinforcement Learning for Solving the Vehicle Routing Problem, arXiv, February 12th, 2018. [Online]. Available: https://arxiv.org/abs/1802.04240.

Fig. 33: Z. Iklassov, I. Sobirov, R. Solozabal, M. Takáč, Reinforcement Learning for Solving Vehicle Routing Problem with Time Windows, arXiv, February 15th, 2024. [Online]. Available: https://arxiv.org/abs/2402.09765

Fig. 34: load_vehicles() flowchart.

Fig. 35: add_vehicles() flowchart.

Fig. 36: step() flowchart.

Fig. 37: episode rewards of the environment before training.

Fig. 38: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Fig. 39: episode's statistics.

Fig. 40: folium illustration with OpenStreetMap, CartoDB (CC BY-SA 2.0).

Tables

Graphs

Acronyms.

Acronym	Meaning
RL	Reinforcement Learning
VRP	Vehicle Routing Problem
ML	Machine Learning
NLP	Natural Language Processing
RTT	Real-Time Tracking
SWOT	Strengths, Weaknesses, Opportunities, Threats
DM	Decision Maker
VRPTW	VRP with Time Windows
TW	Time Windows
CFRS	Cluster First, Route Second
NP	Nondeterministic Polynomial-time
API	Application Programming Interface
ORS	OpenRouteService
ACO	Ant Colony Optimization
VRPSDP	VRP with Simultaneous Delivery and Pickup
ABC	Artificial Bee Colony
HHC	Home Healthcare
PSO	Particle Swarm Optimization
R&R	Ruin-and-Recreate
TAM	Two-stage Divide Method
AI	Artificial Intelligence
MDPs	Markov Decision Processes
RNN	Recurrent Neural Network
SVRP	Stochastic VRP
DQN	Deep Q-Networks
A2C	Advantage Actor Critic
PPO	Proximal Policy Optimization
HOOF	Hyperparameter Optimisation on the Fly
CNN	Convolutional Neural Network
SCM	Supply Chain Management
TNC	Transportation Network Companies
CBA	Cost and Benefit Analysis
ROI	Return On Investment
NPV	Net Present Value
B-CR	Benefit-Cost Ratio

Literature.

[1] C. Hashemi-Pour, J.M. Carew, Reinforcement Learning definition, *TechTarget*, August 2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning

[2] W. Kenton, M. James, V. Velasquez, Logistics: What It Means and How Businesses Use It, *Investopedia*, June 12th, 2023. [Online]. Available: https://www.investopedia.com/terms/l/logistics.asp

[3] About FedEx, *Forbes*, no available date. [Online]. Available: https://www.forbes.com/companies/fedex/?sh=339ff7905638

[4] FedEx, *Wikipedia, The Free Encyclopedia*, January 5th, 2024 (last revision). [Online]. Available: https://en.wikipedia.org/wiki/FedEx#:~:text=FedEx%20today%20is%20best%20known,delivery%20as%20a%20flagship%20service.

[5] A.Z. Firdaus, Solving Vehicle Routing Problems with Python & Heuristics Algorithm, *Medium*, July 21st, 2023. [Online]. Available: https://medium.com/@writingforara/solving-vehicle-routing-problems-with-python-heuristics-algorithm-2cc57fe7079c

[6] Johar, Farhana & Potts, Chris & Bennell, Julia, 2015. Vehicle routing problem with time constraints. Malaysian Journal of Fundamental and Applied Sciences. 11. [Online]. DOI: 10.11113/mjfas.v11n4.400.

[7] F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, M. Yuan, March 1st, 2023. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. [Online]. Available: https://arxiv.org/abs/2303.04147v1

[8] S. Naseera, P, NP, NP-hard, NP-complete problems, *Department of CSE JNTUACEK, Kalikiri*, January 5th, 2023. [Online]. Available: https://www.jntua.ac.in/gate-online-classes/registration/downloads/material/a159262902029.pdf

[9] Fedex facilities, *HIFLD Open Data*. [Online]. Available: https://hifld-geoplatform.opendata.arcgis.com/datasets/fedex-facilities/explore?location=33.473057%2C-89.589712%2C4.42 (accessed Jan. 26, 2024).

[10] F. Holland, FedEx gets first of 500 electric trucks from GM's EV unit in a major advance for green logistics, *CNBC*, December 17th, 2021. [Online]. Available: https://www.cnbc.com/2021/12/17/fedex-gets-first-of-500-electric-trucks-from-gms-ev-unit-in-move-to-green-logistics.html

[11] Planning & Urban Design, Dallas by Numbers 2017, *Dallas City Hall*. [Online]. Available: https://dallascityhall.com/departments/pnv/Pages/Dallas-by-Numbers-2017.aspx (accessed Jan. 26, 2024).

[12] R. Baldwin, What it's like to drive GM's BrightDrop Zevo 600 electric delivery van, *Autoblog*, May 30th, 2022. [Online]. Available: https://www.autoblog.com/2022/05/30/brightdrop-zevo600-review/?guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAN6wPcmLp5_iUVxZsmR3g3inygBfr3t_HvBLdgrhBCeaqca8H1laY-xUHHbaJCWZxLLN8WXB1_ZvP3z1Jh74oE5QoQQX7FgVBw-Ccx-

nT3hkKSFZiwg9PjeuZIRiJON9duo_8yXUqq-9CyT0mJtLn6xweDJNQmL3YHJCRiF61ZDT&guccounter=2

[13] Models - Operations research models and methods. [Online]. Available: https://utw11041.utweb.utexas.edu/ORMM/models/unit/linear/subunits/terminology/index.html#:~:text=Decision%20variables%20describe%20the%20quantities,values%20with%20an%20optimization%20method (accessed Jan. 29, 2024).

[14] A. Aggarwal, Techniques for Subtour Elimination in Traveling Salesman Problem: Theory and Implementation in Python, *Medium*, December 6th, 2020. [Online]. Available: https://medium.com/swlh/techniques-for-subtour-elimination-in-traveling-salesman-problem-theory-and-implementation-in-71942e0baf0c

[15] Informazioni su OR-tools – google for developers, *Google*. [Online]. Available: https://developers.google.com/optimization/introduction?hl=it (accessed Jan. 30, 2024).

[16] OpenRouteService. [Online]. Available: https://openrouteservice.org/ (accessed Jan. 31, 2024).

[17] J. Juviler, What is an API Key? (And are they secure?), *HubSpot*, August 15th, 2023. [Online]. Available: https://blog.hubspot.com/website/api-keys

[18] Unixtimestamp. [Online]. Available: https://www.unixtimestamp.com/ (accessed Jan. 31, 2024).

[19] Keleno, OR-TOOLS, *Medium*, August 26th, 2021. [Online]. Available: https://medium.com/keleno/or-tools-fc7f9a6f59b0

[20] OpenStreetMap Routing and Directions: Pros And Cons, *Geoapify*, January 18th, 2022. [Online]. Available: https://www.geoapify.com/openstreetmap-routing

[21] Marshall Fisher, Chapter 1 Vehicle routing, Handbooks in Operations Research and Management Science, Elsevier, Volume 8, 1995, Pages 1-33. [Online]. Available: https://doi.org/10.1016/S0927-0507(05)80105-7 (accessed Jan. 31, 2024).

[22] N.A. El-Sherbeny, Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods, *Sciencedirect*, April 7th, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1018364710000297#bib4

[23] Complexica, Local search, *Narrow AI Glossary*. [Online]. Available: https://www.complexica.com/narrow-ai-glossary/local-search#:~:text=Local%20search%3A%20Local%20search%20is,visibility%20in%20their%20local%20markets (accessed Feb. 02, 2024).

[24] Metaheuristic, *Autoblocks*. [Online]. Available: https://www.autoblocks.ai/glossary/metaheuristic (accessed Feb. 02, 2024).

[25] Laporte, Gilbert & Ropke, Stefan & Vidal, Chapter 4: Heuristics for the Vehicle Routing Problem, *Thibaut*. (2014). [Online]. Available: https://www.researchgate.net/publication/285833854_Chapter_4_Heuristics_for_the_Vehicle_Routing_Problem

[26] H. Rhim, Tabu Search, *Baeldung*, May 2nd, 2023. [Online]. Available: https://www.baeldung.com/cs/tabu-search

[27] F. Simsir, D. Ekmekci, A metaheuristic solution approach to capacitied vehicle routing and network optimization, *Sciencedirect*, January 23rd, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098618320962

[28] P. Saksuriya, C. Likasiri, Hybrid Heuristic for Vehicle Routing Problem with Time Windows and Compatibility Constraints in Home Healthcare System, *MDPI*, June 26th, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/13/6486

[29] C. Chen, E. Demir, Y. Huang, An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots, *Sciencedirect*, February 13th, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S037722172100120X

[30] N. R. Sabar, A. Bhaskar, E. Chung, A. Turky, A. Song, A self-adaptive evolutionary algorithm for dynamic vehicle routing problems with traffic congestion, *Sciencedirect*, October 31st, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2210650218303407

[31] Q. Hou, J. Yang, Y. Su, X. Wang, Y.Deng, Generalize Learned Heuristics to Solve Large-scale Vehicle Routing Problems in Real-time, *OpenReview.net*, February 1st, 2023. [Online]. Available: https://openreview.net/forum?id=6ZajpxqTlQ

[32] What is Reinforcement Learning, *Synopsys*. [Online]. Available: https://www.synopsys.com/ai/what-is-reinforcement-learning.html (accessed March 4, 2024)

[33] Warwick Data Science Society, Reinforcement Learning in Action, Part 1: Electric Vehicle Routing, *Medium*, July 11th, 2023. [Online]. Available: https://medium.com/@WDSS/reinforcement-learning-in-action-part-1-electric-vehicle-routing-39e70569d7d6

[34] Vijay Kanade, What Is the Markov Decision Process? Definition, Working, and Examples, *Spiceworks*, December 20th, 2022. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-markov-decision-process/

[35] Navneet Singh, Understanding Q-Learning: A Powerful Reinforcement Learning Technique, *Medium*, July 11th, 2023. [Online]. Available: https://medium.com/@navneetskahlon/understanding-q-learning-a-powerful-reinforcement-learning-technique-29a3da36f611

[36] Cambridge Dictionary. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/stochastic (accessed March 14, 2024).

[37] Eindhoven Reinforcement Learning Seminar, Reza Nazari "Reinforcement Learning for Solving the Vehicle Routing Problem", YouTube, March 06, 2021 [Video file]. Available: https://www.youtube.com/watch?v=SNcZAt_vbkY (accessed March 14, 2024).

[38] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, Martin Takáč, Reinforcement Learning for Solving the Vehicle Routing Problem, *arXiv*, February 12th, 2018. [Online]. Available: https://arxiv.org/abs/1802.04240

[39] Z. Iklassov, I. Sobirov, R. Solozabal, M. Takáč, Reinforcement Learning for Solving Vehicle Routing Problem with Time Windows, *arXiv*, February 15th, 2024. [Online]. Available: https://arxiv.org/abs/2402.09765

[40] A. de Wispelaere, Leveraging AI for Route Optimization: Pros, Limits, and Risks, *ptvlogistics*, June 15th, 2023. [Online]. Available: https://blog.ptvlogistics.com/en/transport-logistics/leveraging-ai-for-route-optimization-pros-limits-and-risks/

[41] SaturnCloud. [Online]. Available: https://saturncloud.io/glossary/reinforcement-learning-environments/#:~:text=A%20Reinforcement%20Learning%20Environment%20is,states%20based%20on%20those%20actions. (accessed April 12th, 2024).

[42] C. M. Bowyer, What is State in Reinforcement Learning?, *LinkedIn*, November 3rd, 2022. [Online]. Available: https://www.linkedin.com/pulse/what-state-reinforcement-learning-caleb-m-bowyer/

[43] Team Datatonic, Reinforcement Learning: How to Train an RL Agent from Scratch, *Medium*, October 4th, 2022. [Online]. Available: https://blog.montrealanalytics.com/reinforcement-learning-how-to-train-an-rl-agent-from-scratch-96af39a3287

[44] LinkedIn community, How do you evaluate the performance and robustness of your reinforcement learning agent?, *LinkedIn*. [Online]. Available: https://www.linkedin.com/advice/0/how-do-you-evaluate-performance-robustness-your-reinforcement#:~:text=One%20of%20the%20simplest%20and,compare%20different%20algorithms%20or%20parameters. (accessed April 18th, 2024).

[45] B. Venkatapathy, Evaluation metrics for reinforcement algorithms, *Medium*, November 25th, 2023. [Online]. Available: https://medium.com/@barathchandarcse/evaluation-metrics-for-reinforcement-algorithms-ff2bf5869fe4

[46] Actor-Critic Algorithm in Reinforcement Learning, *Geeksforgeeks*, March 22nd, 2024. [Online]. Available: https://www.geeksforgeeks.org/actor-critic-algorithm-in-reinforcement-learning/

[47] D. Kumar, PPO Algorithm, *Medium*, February 21st, 2024. [Online]. Available: https://medium.com/@danushidk507/ppo-algorithm-3b33195de14a

[48] S.Paul, V. Kurin, S. Whiteson, Fast Efficient Hyperparameter Tuning for Policy Gradients, *Deptartment of Computer Science University of Oxford*, Februay 18th, 2019. [Online]. Available: https://arxiv.org/abs/1902.06583

[49] A2C, *Stable Baselines3*. [Online]. Available: https://stable-baselines3.readthedocs.io/en/master/modules/a2c.html

[50] PPO, *Stable Baselines3*. [Online]. Available: https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html

[51] Gymnasium Documentation, *Gymnasium*. [Online]. Available: https://gymnasium.farama.org/index.html

[52] Github repository of @zestyraiden, *Github*. [Online]. Available: https://github.com/zestyraiden/Reinforcement-Learning-VRP/blob/master/Notebooks/VRP%20-%20Reinforcement%20Learning%20.ipynb

[53] M. Fadel Argerich, Entropy Regularization in Reinforcement Learning, *Medium*, May 24th, 2020. [Online]. Available: https://towardsdatascience.com/entropy-regularization-in-reinforcement-learning-a6fa6d7598df

[54] A. Dhinakaran, Understanding KL Divergence, Medium, February 2nd, 2023. [Online]. Available: https://towardsdatascience.com/understanding-kl-divergence-f3ddc8dff254

[55] T. Rochefort-Beaudoin, A. Vadean, N. Aage, S. Achiche, Structural Design Through Reinforcement Learning, *Department of Mechanical Engineering, Polytechnique Montréal*, July 2024. [Online]. Available: https://www.researchgate.net/publication/382145625_Structural_Design_Through_Reinforcement_Learning

[56] AurelianTactics, Understanding PPO Plots in TensorBoard, *Medium*, December 14th, 2018. [Online]. Available: https://medium.com/aureliantactics/understanding-ppo-plots-in-tensorboard-cbc3199b9ba2

[57] T Carr, Deterministic vs. Stochastic Policies in Reinforcement Learning, *Baeldung*, March 18th, 2024. [Online]. Available: https://www.baeldung.com/cs/rl-deterministic-vs-stochastic-policies

[58] S. Dhumne, Deep Q-Network (DQN), *Medium*, June 30th, 2023. [Online]. Available: https://medium.com/@shruti.dhumne/deep-q-network-dqn-90e1a8799871

[59] ETICA, What is the significance of the exploration-exploitation trade-off in reinforcement learning?, *Etica*, May 13th, 2024. [Online]. Available: https://eitca.org/artificial-intelligence/eitc-ai-arl-advanced-reinforcement-learning/introduction-eitc-ai-arl-advanced-reinforcement-learning/introduction-to-reinforcement-learning/examination-review-introduction-to-reinforcement-learning/what-is-the-significance-of-the-exploration-exploitation-trade-off-in-reinforcement-learning/

[60] U. Tewari, Regularization — Understanding L1 and L2 regularization for Deep Learning, *Medium*, November 9th, 2021. [Online]. Available: https://medium.com/analytics-vidhya/regularization-understanding-l1-and-l2-regularization-for-deep-learning-a7b9e4a409bf

[61] S. Karagiannakos, Regularization techniques for training deep neural networks, *The AI Summer*, May 27th, 2021. [Online]. Available: https://theaisummer.com/regularization/

[62] Z. Liu, X. Li, Bi. Kang, T. Darrell, Regularizations in Policy Optimization - An Empirical Study on Continuous Control, *International Conference on Learning Representations*, January 12th, 2021. [Online]. Available: https://openreview.net/forum?id=yr1mzrH3IC

[63] S. Fo`, C. Coppolaa, G. Granib, L. Palagia, Solving the vehicle routing problem with deep reinforcement learning, *Sapienza University of Rome*, July 30th, 2022. [Online]. Available: https://arxiv.org/pdf/2208.00202

[64] TensorBoard. [Online]. Available: https://www.tensorflow.org/tensorboard?hl=it

[65] Stable-Baselines3. [Online]. Available: https://stable-baselines3.readthedocs.io/en/master/index.html

[66] P. Schwartenbeck, A. Zap, Reinforcement Learning Use Cases for Business Applications, *Alexanderthamm*, June 21st, 2023. [Online]. Available: https://www.alexanderthamm.com/en/blog/reinforcement-learning-use-cases-for-companies/

[67] R. Schmelzer, Amazon Dives Deep into Reinforcement Learning, *Forbes*, June 14th, 2019. [Online]. Available: https://www.forbes.com/sites/cognitiveworld/2019/06/14/amazon-dives-deep-into-reinforcement-learning/

[68] D. Shi et al., Deep Q-Network Based Route Scheduling for Transportation Network Company Vehicles, *2018 IEEE Global Communications Conference (GLOBECOM),* Abu Dhabi, United Arab Emirates, 2018, pp. 1-7. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8647546

[69] Geocoding, *Precisely*, No available date. [Online]. Available: https://www.precisely.com/glossary/what-is-geocoding#:~:text=Geocoding%20is%20the%20process%20of,targeted%20mailings%20and%20timely%20deliveries.

[70] A. Hayes, Cost-Benefit Analysis: How It's Used, Pros and Cons, *Investopedia*, July 25th, 2024. [Online]. Available: https://www.investopedia.com/terms/c/cost-benefitanalysis.asp#toc-the-cost-benefit-analysis-process

[71] FAQ Wrike, What Is Net Present Value (NPV) in Project Management?, *Wrike*. [Online]. Available: https://www.wrike.com/project-management-guide/faq/what-is-net-present-value-npv-in-project-management/

[72] A. Athuraliya, A step-by-step Guide to Scenario Planning, *Professional Academy*, No available date. [Online]. Available: https://www.professionalacademy.com/blogs/a-step-by-step-guide-to-scenario-planning/

Acknowledgments.

First of all, I would like to thank my professor Raffaele Pesenti, supervisor of this thesis. He has always been supportive with my work and interested in what I was doing. I will never forget his help and his commitment. In addition, I want to thank my professor Denis M. Becker, co-coordinator of this research: he helped me in the coding part and provided me with interesting material.

My university mates have been the best colleagues I could have ever wished for. Beatrice, Francesco, Gabriele and Marco, you have been true and special friends that I will always bring in my heart. We still have a lot of memories to create together and I hope we stay as close as we are now.

During my Erasmus in Norway, I met amazing people who warmly welcomed me and became friends for life. Carole, with you I spent the funniest moments and I always felt at home in our living room; leaving you has been the hardest part, but I know that the future still wants us together. Maja, you are the revelation of my Norwegian life; our friendship started on tiptoe but now I feel closer than ever to you and I hope we will live more experiences together. Ingrid, your enthusiasm for life and your constant glow: you are one of the most amazing persons I know, don't ever turn off your light. Alessia, my Italian friend with who I shared the deepest moment of my Erasmus: thank you for being like a sister and taking care of me when I was feeling down. I also want to thank Mattia and Michele for being the best project mates in Norway: with you I worked efficiently and I was always happy to share moments with you.

I can't skip an important part of my life, my years at Foot Locker. They have been the most important to me, because I met special people and I understood the value of working in a safe and careful environment. Without the support of Guido, Giulia, Martina P., Martina V., Cosimo, Nicola, Sebastiano, Mahedi, Akash and all the other colleagues, I could have never reached my life goals. From the bottom of my heart, thank you.

Arianna, you are one of those friends that you rarely meet. We have known each other since we were kids: this strong and unbreakable bond that we have is one of the most special things of my life. I wish we lived closer to share more moments together, but every time we meet it's like we never were apart.

Sara, I want to thank life for bringing us together again. Without your constant support and your true friendship, I could have never understood the difference between a genuine and an opportunistic friend. I want to keep making mistakes with you, getting back up from letdowns together and travel to new worlds side by side. Thank you for always believing in our friendship.

Chiara, nobody compares to you. We have been through a lot during the past 18 years and I am so glad we never lost our sparkle, even in the quietest moments of our friendship. We laugh, we cry, we travel, we share passions and most importantly, we are who we are when we are together. I can be my true self with you and I know I will never get judged. Thank you for never letting me go.

I also want to thank Margherita, Jessica and Laura for always being there and for the moments shared together. You are special to me.

Shain, even if we haven't met a lot lately, I want to thank you because since the first year of University, you have never let me go. You have been the best Bachelor's mate and without you I could have never gone through those hard three years.

An important part of my heart lies in Paraguay: there, I spent a wonderful year of my life and I hope to go back there again. My host family has been supporting me since the day we said goodbye and this gave me the strength to go on. You are always with me, even if there is an entire ocean tearing us apart.

Anyway, my first supporter is my family. You raised me with all the resources you had and you taught me to always believe in my dreams and my capabilities. Thank you, because it is not taken for granted. Thank you, because even if we don't live together anymore, I always feel you close to me. Thank you, because you are the best family in the world and I never felt like I didn't have enough. I always had enough. Thank you for teaching me resilience and for always believing in YOUR dreams. Thank you for giving me the opportunity to go to Paraguay, to study, to go to Norway, to work and to always be myself. I love you.

I also want to thank Flavia, Roberto and Carlotta. You welcomed me for two years and made me feel at home, always. You always make me laugh and feel accepted and this is something I will never forget. Thank you for being like a family and treating me like your own daughter and sister.

Leonardo, you are my person. Without you, nothing makes sense. You support me, you love me, you help me, you make me laugh and grow and you never let me down. To me, your love is the most valuable treasure I have in life and I never want to lose it. I would have never understood the meaning of love if I hadn't met you and I hope life has something great in store for us. I want to achieve goals, make a family, live truly and grow old with you. Because it's you, it has always been you. Thank you for these four years and for the ones that will come because I already know, I know they are going to be amazing.