REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Department of Industrial Engineering
Option: Industrial Management

FINAL PROJECT REPORT

In order to obtain the diploma of State Engineer in Industrial Engineering

Optimization of fuel distribution network using Deep Reinforcement Learning
Case of NAFTAL

KHELIFA MAHDJOUBI Nazih

Under the supervision of Dr. Iskander Zouaghi
And
Mr. KAOUANE Director of NAFTAL products optimization

Presented and supported released on (29/06/2022)

Composition of the Jury:

President	Ms. Bahia Bouchafaa (MCA)	ENP
Promoter	Mr. Iskander Zouaghi (MCA)	ENP
Examiner	Ms. Wedjdane Nahili (MCB)	ENP

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Department of Industrial Engineering
Option: Industrial Management

FINAL PROJECT REPORT

In order to obtain the diploma of State Engineer in Industrial Engineering

Optimization of fuel distribution network using Deep Reinforcement Learning
Case of NAFTAL

KHELIFA MAHDJOUBI Nazih

Under the supervision of Dr. Iskander Zouaghi
And
Mr. KAOUANE Director of NAFTAL products optimization

Presented and supported released on (29/06/2022)

Composition of the Jury:

President	Ms. Bahia Bouchafaa (MCA)	ENP
Promoter	Mr. Iskander Zouaghi (MCA)	ENP
Examiner	Ms. Wedjdane Nahili (MCB)	ENP

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département de Génie Industriel
Option : Management Industriel

Mémoire de Projet de Fin d'Etudes
En vue de l'obtention du diplôme d'Ingénieur d'État en Génie Industriel

Optimisation du réseau de distribution de carburant à l'aide du Deep Reinforcement Learning, Case de NAFTAL

KHELIFA MAHDJOUBI Nazih

Sous la direction de Dr. Iskander Zouaghi
Et
Mr. KAOUANE Directeur de l'optimisation des produits NAFTAL

Présenté(e) et soutenue publiquement le (29/06/2022)

Composition of the Jury:

Président	Mme. Bahia Bouchafaa (MCA)	ENP
Promoteur	M. Iskander Zouaghi (MCA)	ENP
Examinateur	Mme. Wedjdane Nahili (MCB)	ENP

ENP 2022

Acknowledgment

First of all, A thanks message to all my family, brothers and sisters...

Djed,
Imad,
Akram ...

Especially my dear mother for all her efforts with me during all my paths in order to reach this moment and everyone, who knew me and helped in my life,

We would like to thank our supervisor, Dr. Iskander Zouaghi, for his interest in our work, his availability to us, as well as his valuable advice. He gave us the benefit of his experience and his scientific background and Mr. KAOUANE Director of NAFTAL products optimization for his remarks and suggestions were a valuable contribution.

We also thank the entire teaching team of the Algiers polytechnic school, especially the teachers of the department of industrial engineering to whom we owe our engineering formation.

We extend our thanks to Ms. Wedjdane Nahili and Ms. Bahia Bouchafaa, for having accepted to examine our work.

Finally, we would like to thank all those who have participated directly or indirectly to the realization of this project.

والحمد لله رب العالمين أولا وأخيرا ...

الملخص
يركز الهدف من هذا العمل على دراسة مشاكل النقل البري الحقيقية من خلال اقتراح النماذج والحلول المتعلقة بتحسين أداء شبكة توزيع محطات الوقود. في هذا العمل ، نركز على تحسين التحكم في الطلب وإدارة المخزون للمحطات باستخدام تقنيات إنترنت الأشياء وأدوات التنبؤ ، وتحسين توجيه المركبات متعددة المقصورات مع مشكلة المدة الزمنية (MCVRPTW)الناشئة في توزيع المنتجات البترولية وصناعة نقل الحاويات، مع ثلاثة مناهج مختلفة باستخدام النموذج الرياضي ، ونهج التعلم الاستكشافي (الأدلة العليا) والتعلم عن طريق التعزيز العميق.

الكلمات الرئيسية: التحسين، التوزيع، إنترنت الأشياء، التوقع الاحتمالي، التحسين بالاستمثال التوافقي العصبي، الأدلة العليا، اللوجيستيات البترولية .

Résumé
L'objectif de ce travail se concentre sur l'étude de problèmes réels de transport routier en proposant des modèles et des solutions liés à l'amélioration de la performance du réseau de distribution des stations-service. Dans ce travail, nous nous concentrons sur l'optimisation du contrôle de la demande et de la gestion des stocks de la station en utilisant les technologies IoT et les outils de prévision, et nous améliorons le problème de routage de véhicules multi-compartiments avec fenêtres de temps (MCVRPTW) qui se pose dans l'industrie de la distribution de produits pétroliers et du transfert de conteneurs, avec trois approches différentes utilisant un modèle mathématique, une approche heuristique et une approche d'apprentissage par renforcement profond.

Mots-clés : optimisation, distribution, IoT, prévision, optimisation combinatoire neuronale, métaheuristique, DRL, VRP, logistique pétrolière.

Abstract
The goal of this work focuses on the study of real road transportation problems by proposing models and solutions related to the improvement of performing the gas station distribution network. In this work, we focus on the optimization of demand control and Inventory management of station using IoT technologies and forecasting tools, and improve the multi-compartment vehicle routing with time windows (MCVRPTW) problem arising in the petroleum products distribution and container transfer industry, with three different approach using mathematical model, heuristic and Deep Reinforcement Learning approaches.

Keywords: Optimization, distribution, IoT, Forecasting, neural combinatorial optimization, metaheuristics, DRL, VRP, petroleum logistics

Table of Contents

Tables list

List of the figures

List of Acronyms

Table of Contents

Tables list

List of the figures

List of Acronyms

ACF: Auto Correlation Function
AR: Auto Regressive
ARIMA: Auto Regressive Integrated Moving Average
ARMA: Auto Regressive Moving Average
ATG: Automatic Tank Gauging
DA: Algerian Dinars
DS: Difference stationary
DRL: Deep Reinforcement Learning
ESCO: Energy Service Company
FIFO: First In First Out
FRC: Form for Receipt of Order
GD: Direct Management
LPG: Liquefied Petroleum Gas
IoT: Internet of Things
MA : Moving Average
MAPE : Mean Absolute Percentage Error
MDP: Markov Decision Process
OLS: Ordinary Least Squares
PVA : Point of Sale
RFID: Radio Frequency Identification
SARIMA: Seasonal Auto Regressive Integrated Moving Average
SCM: Supply Chain Management
SONATRACH: National company for research, production, transport, processing and marketing of hydrocarbons marketing of hydrocarbons
SS: Safety Stock
TR: Reality Rate
TS: Trend Stationary

General introduction

The petroleum industry is a key element of the global industrial infrastructure, where the upstream part of the industry is the extraction of oil. In opposite, the downstream part is composed of the refining, sales and distribution sectors. The hydrocarbon sector represents one of the most important industrial sectors in Algeria. Indeed, the Algerian economy remains strongly dependent on it since this sector represents the main source of income for the country. In addition, being a producer and exporter of fuel, it must take charge of local and foreign demand. In Algeria, the distribution and marketing of petroleum products are the responsibility of NAFTAL.

The supplier, usually in a monopoly or oligopoly situation, is responsible for replenishing a large number of service stations. Some of these are managed by the supplier in question (GD service stations), while others are managed by private operators (PVA Stations). This makes the logistics of this supplier more and more complex and difficult to manage and consequently impacts the performance of service stations. The final part of the petroleum chain is the distribution of the finished products to the points of sale, using three modes of transport, which vary according to the region: the sea, the railroad and the road network, it is by this last mode of transport that the service stations are supplied from intermediate depots or directly from the oil refineries. In recent years, the managers of big companies including NAFTAL have been facing many complex problems, and this is due to the variation in the volume of sales in addition to the increasing demand and can be led to a failure to meet the demand. Of which it is important to find methods and mathematical techniques to manage the function of the supply chain activity.

These approaches and modern scientific concepts that have emerged to respond to the demand and logistics as an integrated system, in other words, in order to ensure that its products will be delivered to the market, it must design a complete system to control the flow of materials and the requirements of production, and its products flow, a possible solution would be to develop a model that will allow us to better control and optimize the chain's flows, adapted to the actual context in which it evolves. In view of the constraints, it is important for the fuel station to control the internal and external parameters by adapting an efficient decision making system that will allow determining the deviations between actual and optimal parameters, in our case we analyze and explain them in order to improve the performance of NAFTAL distribution network. To do this, our following work was structured in three chapters as follows:

The first is the presentation of the company where the project was done, namely "NAFTAL", which had as objective “the optimization of the performance of the stations and the fuel distribution network”. We will therefore present this company, then, we will develop a diagnosis to detect possible research areas of improvement of the network distribution to the service station.

The second chapter allows us to identify the theoretical framework of our study, which will concern the main concepts of the fuel supply chain as well as the necessary tools to optimize the chain, namely the use of forecasting tools, demand planning, as well as the neural combinatorial optimization using transformers for decision making.

As for the third chapter, it will be focused, on the one hand, on the resolution of the internal operational problems that the service station is facing, namely, stock-outs, over-stocking, by developing a data entry framework using IoT tools and on the other hand, on the optimization problems of the downstream supply chain that the service station distributor is facing with two different approaches (classical heuristics and innovative deep reinforcement learning approach).

Finally, a general conclusion will summarize our work and present the different perspectives in order to anticipate the possible future projects.

Chapter 01: Presentation of NAFTAL and the background project

1 Introduction

Firstly, we start in this chapter by the presentation of the company where the project was carried out, for reason to fully understand the industrial context of our chapter; we begin by presenting a general overview of its field of application. Thus, throughout this section, we present the essential elements that can give a presentation of the NAFTAL Company. These elements relate to the history of the company, its main missions and the various aspects of the organization of fuel distribution and its challenges in Algeria and finishing by describing various problematic (the research question) and to increase the company distribution and the station performance in our project.

2 Presentation of the company

NAFTAL is an Algerian company, a 100% of SONATRACH with a capital of 160.000.000.000 DA.[1] It is in charge of the distribution of petroleum products on the Algerian and Tunisian markets. It is specialized in the conception, the elaboration and the distribution of lubricants for engines (two-wheelers, cars and other vehicles).

Created by the decree N°80101 of April 6, 1980, the company "ERDP-NAFTAL" started its activity on January 1, 1982. It is in charge of the refining industry and the distribution of petroleum products. Before this day, the entirely distribution and marketing of petroleum products in Algeria depended on large multinational companies such as "SHELL", and "BRITISH PETROLIUM".

In 1987, the refining activity is separated from the distribution activity by the decree No. 87199 of August 27 in two companies:

- NAFTEC: responsible for refining oil.
- NAFTAL: in charge of the distribution and marketing of petroleum products.

NAFTAL Missions

The marketing, storage, transport and distribution of petroleum products and derivatives, in particular, fuels and lubricants, including those intended for aviation and the navy, LPG, solvents, aromatics … The coverage of the needs of the national market in petroleum products and derivatives by:

- Organize and develop the marketing and distribution of petroleum products and derivatives.

[1] Source: NAFTAL internal source

- Store and transport all petroleum products marketed on the national territory.

The development of all forms of joint activities in Algeria and outside Algeria with Algerian or foreign companies, the taking and the holding of all portfolios of shares, the taking of participations and other securities in all existing or to be created companies in Algeria or abroad;

- To ensure the application and respect of measures relating to industrial safety, the safeguarding and protection of the environment.
- To carry out any market study of petroleum products consumption.
- To define and develop an audit policy.
- Design and implement information systems.
- Develop and implement actions aiming at an optimal and rational use of means and rational use of means and infrastructures.

Ensure the application and respect of measures related to the internal security of the company, in accordance with the regulations, and generally, all industrial, commercial, financial, movable or immovable operations that can be directly or indirectly related to its corporate purpose. In network of 2,276 service stations (692 SS in NAFTAL ownership) and 3,600 units transport fleet.

NAFTAL's Organization

The organization of NAFTAL illustrated in the flow chart in Figure 1 is as follows:

- **General direction:** In charge of policy and general orientations, coordination and coherence of the coordination and coherence of the steering group. It is ensured by a president

Director General assisted by:

- An executive committee.
- A director's committee.
- Advisers.

- **Executive Direction:**

Each one in its field of activity, are in charge of:

- Defining the company's policy and strategy.
- Anticipating trends.
- Designing and implementing steering and control tools.
- Ensuring and assisting in the coordination and coherence of the group.

- Ensure the operational structures (marketing branch, fuel branch and LPG branch).

– **Central direction:** It is a center of expertise (estimation) for marketing, research and development and auditing activities (analysis and control of management and accounting of the company)

– **Direction of supports:** Provides administrative management of the company's headquarters.

The different operational structures are based on three main branches:

- Fuel branch: It is made up of 10 districts (composed of a defined number of fuels of fuel depots), aviation depots and marine centers.
- Marketing branch: It is made up of 12 districts (comprising a defined number of distribution centers, lubricant/tire centers and a network of service stations).
- LPG Branch: It is made up of 19 districts regrouping a defined number of bulk centers, filling centers and relay depots.

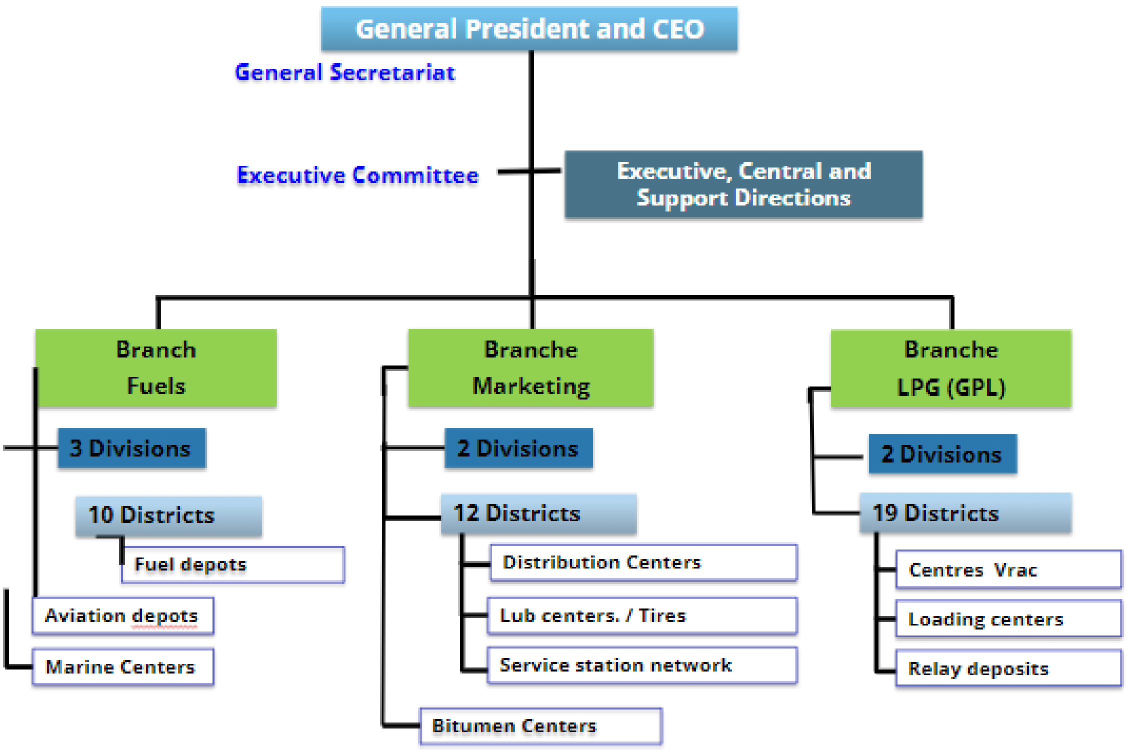


Figure 1 NAFTAL Fuel branch architecture

Fuel branch presentation

Our study will be limited to the ground fuel district level. The missions and tasks of the latter are to:

- Direct and control the fuel depots and ensure the maintenance of pipelines and rolling stock.
- Manage the "supply" and "refueling" flows of the product to the depots of the depots to which it is attached.

In addition to the tasks mentioned, some districts like Bechar, Ouargla and Batna, are in charge of the distribution activity.
As for the fuel centers, the head of the fuel center, under the authority of his district are responsible for:

- Managing, controlling and measuring the flows, in quantity and quality, of fuel products, to and from his center Manage, control and measure the quantity and quality of fuel products to and from his center;
- Ensuring the maintenance and industrial safety of the installations and other resources of the of the center;
- Establish and control the accounting day

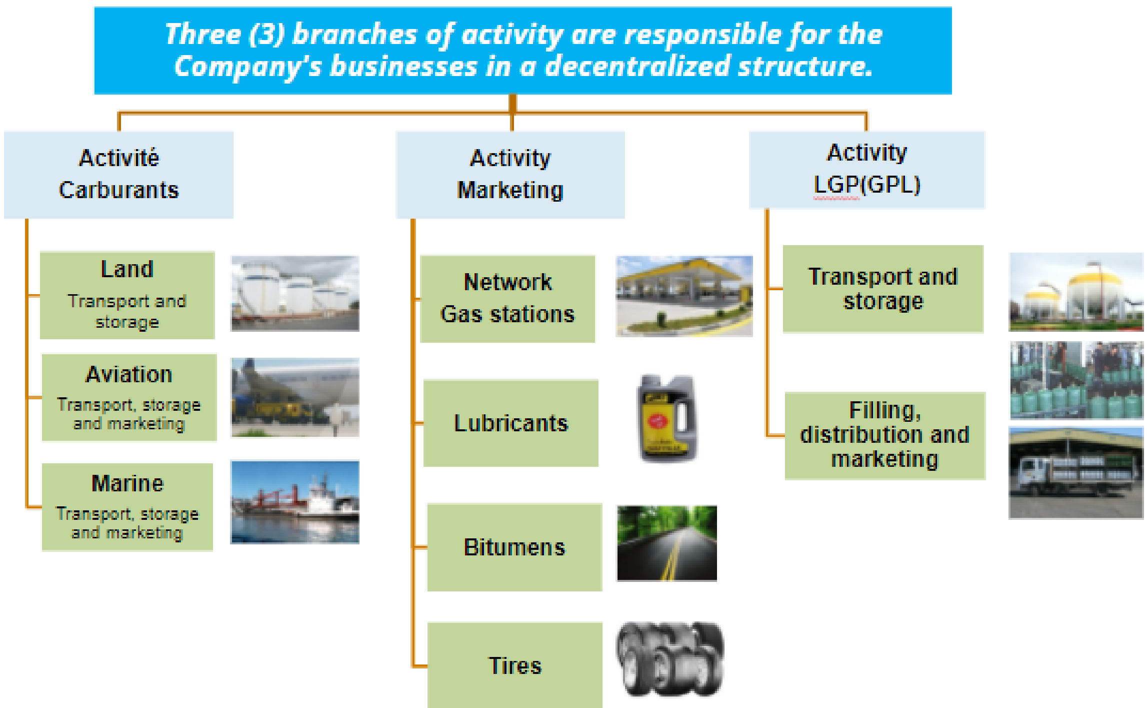


Figure 2 NAFTAL organisation chart

Distribution policy: NAFTAL, is a company of distribution and commercialization of petroleum products, its objective is to bring its products to its customers by its own or private means. NAFTAL clientele is classified into three main categories:

- Directly Managed (DM): these are stations managed directly by NAFTAL with the help of salaried employees.
- Self-managed (SM): these are stations owned by NAFTAL whose business is leased to private individuals.
- Licensed Sales Stations (VAS): service stations operated entirely by investors.

Also there are consumers with large storage capacities "big client" like public and private companies, Local governments and communities, Hospitals...

Order processing: dived into three phases as follows:

I. **Receipt of order:**

This operation begins each day at 8:00 am and ends at 1:00 pm. The reception is done by telephone, fax, and purchase order or by the presence of customers in NAFTAL. After receiving the orders, they are recorded in an ORF form (order reception form). This form contains information concerning the customers, like (name, first name, customer code, quantity requested, date of order, name of carrier, name of station). The satisfaction of the orders is done one day later (D+1), that is to say, the customers make their order and receive them the following day.

II. **Dispatching**:

After the orders are received, the dispatching phase begins. In this phase the ORF form is sent to the dispatcher (the person who gives the distribution plan of the products), to elaborate an optimal distribution plan.
To do this, the following points must be taken into consideration

- The location of the stations and their capacities.
- The availability of drivers and trucks.
- The availability of products (the quantity in stock).
- Compartments of each available truck.

III. **Billing:**

After establishing the distribution plan, a copy of the ORF form is sent to the scheduling station, where an output slip is sent in turn to the billing station to obtain a delivery slip.

At the billing level, the operations are done with the dispatching software that provides a form consisting of two parts:

- Billing part: It gathers the information concerning the customer.
- Cash box part: It is the part where the customer pays his invoice.

3 Project background

Since the service station's main activity is the sale of fuel, the availability of fuel is essential to the operation of the service station with the mono source of replenishment, so it depends entirely on the organization of the distributor network. The service station must therefore be reactive to customer demand through product availability and the only way to act on this availability is the optimization of its internal operations, in other words, an intervention on the control of the demand by a reliable forecast, an optimal management of stocks and an elimination of the losses causes.

3.1 Stations problem

The dysfunctions which the service station is faced are linked to its internal organization on the one hand, i.e. (the poor control of demand, stock management, and the stoppages in the activity due to replenishment), and to the poor external organization linked to the distributor on the other hand.

To estimate the total costs incurred by the service station, we will first introduce the calculation of costs related to poorly controlled stocks (stock outs and overstocking), then the costs related to the stoppages of the station due to replenishment. To do this, we introduce the following notations:

$Creap$: Non-direct costs associated with the service station's out-of-hours shutdown (DA);
$Crupt$: Non-direct costs generated by stock outs (DA) ;
$Csurst$: Direct costs generated by penalties due to overstocking (DA) ;
Tc: Consumption rate per hour (Liter/hour) ;
N: The average number of hours the station is out of service (hour);
Pd: Unit selling price of fuel (Da/Liter) ;
Nhr: Number of hours out of stock (hour);
$Nsurp$: Number of liters in overstock (Liters) ;
PL: Unit cost generated by the penalty (Da/Liter).

$$Crupt = Tc * Nhr * Pd; \ (1)$$
$$Creap = Tc * N * Pd; \ (2)$$
$$Csurst = Nsurp * PL; \ (3)$$

Numerical application: After determining the dysfunctions of the service station, we calculated the amount of the costs generated by the out-of-stock situations, the over-stockings as well as the costs related to the stops of the activity of the Garidi station (KECHIDI, 2019) from 21/01/2019 to 21/02/2019, and the results are as follows:
The total cost incurred by the service station during the month of January is:

$$CT = Créap + Crupt + Csurst;$$

$$CT = 1{,}217{,}280.903 + 112{,}035.487 + 90{,}579 ;$$

We can deduce that the losses represent 5.6% of the Garidi station total revenue.

$$CT = 1{,}419{,}895.39 \text{ Da}$$

So, the objective is to reduce or even eliminate losses made by the service station, by acting on the control of the demand on the one hand and on the instability of the supplier on the other hand. And this, by establishing the necessary security measures concerning the availability of the product in quantity and at the right time. How can we improve the performance of service station management in terms of reactivity? How can we identify demand and anticipate the management of flows? And finally, how can we stabilize fuel delivery times from the distributor to the service station? We will answer these problems and questions in the coming chapters.

3.2 Fuel network distributions problem

In addition to its activities of distribution and marketing of petroleum products on the national market, the company NAFTAL is also in charge of the daily planning of supply service stations. At the level of the commercial branch of NAFTAL, the preparation of a purchase order is done using specific codes of the above-mentioned information except the nature of payment. After this task, the orders (represented by the sheets) go to the confirmation stage to check the possibility of satisfying them according to the availability of products at the stock level.

The next task is the billing stage where a bill is issued for each confirmed order. As soon as the orders are billed, they are forwarded to the dispatcher who takes into account the characteristics of each tanker, the products transported and the route to be taken. Therefore, the design of the routes for the tanker trucks is done in order to satisfy the customer orders.

"The problem is to determine the trips that the vehicles have to take to the service stations (sale points)."

This is done by road from depots and takes into account the constraints specific to the transport of fuels (material, temporal, regulatory constraints, etc.). For each of its depots throughout the country, NAFTAL has a fleet of tank trucks, including compartments that can contain only one type of product per round. The trucks are limited by their capacity and are not equipped with flow meters, which mean that the entire contents of one or more compartments have to be emptied when a station is delivered. The service stations can order one or more types of products. Moreover, each of these stations imposes a time window within which it wants to receive its delivery. The distribution of fuels at NAFTAL is ensured by the “Dispatching function” which prepares a distribution program.

Each day, the distribution center receives customer orders,; a reception agent prepares a form for each order. The form contains all the necessary information which is:

- The products ordered.
- The quantity ordered of each product.
- The identity of the customer (name of the customer (person or company)) and address.
- Type of payment.
- Time and date.

The goal in this structure is the preparation of a daily distribution program for the fleet at the disposal of the center, in order to satisfy the customer orders. This program consists of a set of rounds (rotations) to be built for each tanker of the fleet. It is of the form:

Tankers	Rotation 1 (Start of service)		Rotation 2 (Start of service)			Rotation 3 (Start of service)
Tankers ‘i’	Depot 1		Depot 2			Depot 3
	Delivery1	Delivery2	Delivery3	Delivery4	Delivery5	Delivery6

Table 1 Daily dispatching schdule

A rotation for a given tank is defined by:

- The depot number for its shipment,
- The time of arrival at the depot,
- All the customer deliveries to be made in this rotation. A delivery corresponds to a customer order or a part of this order.

Related works

The problem, as defined above, has its roots in the multi-compartment time-window vehicle tour problem MCVRP-TW. This problem, despite its many applications in logistics, is little studied in the literature. Multi-compartment vehicle routing problems (MCVRP) were born (Brown & Graves, 1981), which consider multiple compartments in a vehicle to serve more than one type of product at the same time. The optimal solution can be found within acceptable time when the problem scale is small. The exact algorithms for solving MCVRP with Single depot mainly include branch-and-price, branch-and-cut, and branch-price-and-cut methods. (Archetti, Campbell, and Speranza , 2014) compared strategies for handling separately and jointly of distribution of MCVRP.

Since MCVRP was originally proposed for the fuel distribution application, many studies have been conducted in this area. (Cornillier, Boctor, Laporte, & Renaud, 2008); (Cornillier, Laporte, Boctor, & Renaud, 2009); (Popović, Vidović, & Radivojević, 2012); (Vidović, M., Popović, D., & Ratković, B., 2014). The MCVRP in other applications have also been studied, such as recycling and waste management (Elbek, M., & Wøhlk, S., 2016); (Rabbani, M., Farrokhi-asl, H., & Rafiei, H., 2016), olive oil collection (Lahyani, R., Coelho, L. C., Khemakhem, M., Laporte, G., & Semet, F., 2015).

In the joint situation, there are ”no-split and split problems” depending on whether a customer's multiple products must be delivered by one vehicle or not. A mixed integer linear programming model for the split problem was presented, and a branch-and-cut algorithm was developed as the solution method. (Archetti, Bianchessi, and Speranza , 2015) formulated a set partitioning model by making use of an exponential number of variables for the split MCVRP, developed a branch-price-and-cut solution approach. (Thirty years of inventory routing. , 2015) introduced four main categories of MCVRP according to the split or no-split situation for compartments and tanks. Two mixed-integer linear programming formulations were proposed for each case, and specialized models were also suggested for some particular versions of the problem. It was indicated that a branch-and-cut algorithm is applicable to all variants of MCVRP. (Erratum to: A branch-and-price algorithm for two multi-compartment vehicle routing problems. , 2017) presented the no-split and split versions of MCVRP, and presented a Branch-and-Price algorithm for solving to optimality and compared the optimal costs of the two versions.

There are only few articles that have tried to study the MCVRP-TW. Let us also mention the two papers by (Melechovský, 2013). In the first paper, he proposed a heuristic approach based on local search. In the second one, (A genetic algorithm-based classification approach for multicriteria ABC analysis., 2018) were largely inspired by work of Melechovský's and approached the MCVRP-TW with profit. The authors proposed a hybrid approach combining the genetic algorithm and iterated local search.

We claim that our work in MCVRP is the first in this field using Deep Reinforcement Learning. We were inspired by this idea through a group of scientific papers in this field like (Li, J., Xin, L., Cao, Z., 2021) for Solving VR Problem via DRL, in order to reach a significant solution in this field. And (Xin, L., Song, W. Al, 2021) use deep learning for Solving Routing Problems and (Bo Peng) use a Dynamic Attention Model for Vehicle Routing Problems, also (Jingwen Li, Yining Ma, Ruize Gao) use a DRL algorithm for Solving the Heterogeneous Capacitated Vehicle Routing Problem.

Chapter 02 State of Art

1 Introduction

The objective of our project is to optimize the supply chain of NAFTAL by acting on the various parameters that characterize it. These parameters mainly concern minimizing the lost costs of the station by controlling the demand, optimizing the stock management and finally optimizing the supply chain, and developing a fuel distribution model. To do this, we will firstly present supply chain concepts then the different notions and methods allowing the optimization of each of the parameters characterizing it, namely the methods of predictive analytics, and inventory management using IoT and finally, the tools of optimization of the fuel distribution.

2 Supply chain in petroleum industry

A supply chain is the network of all the individuals, organizations, resources, activities and technology involved in the creation and sale of a product. A supply chain encompasses everything from the delivery of source materials from the supplier to the manufacturer through to its eventual delivery to the end user. The supply chain segment involved with getting the finished product from the manufacturer to the consumer is known as the distribution channel.

Oil is one of the world's most important raw materials. It has been the world's leading source of energy since the mid 1950's. The oil industry is one of the most important components of the world economy and has a significant impact on the development of other industries. This energy source is what fuels cars, provides electricity to heat homes and water, is used in modern medicine, processes extract the chemicals used for household cleaning products, and much more. The oil and gas industry play a critical role in driving the global economy. The products that this industry makes support many other vital industries like the automotive industry and manufacturing industry.

Changes in technology, markets and customer needs affect the competitiveness of companies, which requires continuous restructuring of the strategy and tactics of positioning the oil business. Currently, the main problem facing the oil industry is to minimize the cost of production and supply of finished products to consumers. Effective supply chain management can increase the efficiency and competitiveness of a petrochemical plant and its supply as a whole. In a supply-chain, a company is linked to its upstream suppliers and downstream distributors as materials, information, and capital flow through the supply-chain (Lisitsa,2019).

The supply chain of the petroleum industry is extremely complex compared to other industries. It is divided into two different, yet closely related, major segments: the upstream and downstream supply chains. The upstream supply chain involves the acquisition of crude oil, which is the specialty of the oil companies. The upstream process includes the exploration, forecasting, production, and logistics management of delivering crude oil from remotely located oil wells to refineries.

The downstream supply chain starts at the refinery, where the crude oil is manufactured into the consumable products that are the specialty of refineries and petrochemical companies. The downstream supply chain involves the process of forecasting, production, and the logistics management of delivering the crude oil derivatives to customers around the globe. Challenges and opportunities exist now in both the upstream and downstream supply chains.

The logistics network in the petroleum industry is highly inflexible, which arises from the production capabilities of crude oil suppliers, long transportation lead times, and the limitations of modes of transportation. Every point in the network, therefore, represents a major challenge (Managing inflexible supply chains., 1998). The oil and petrochemical industries are global in nature. As a result, these commodities and products are transferred between locations that are in many cases continents apart. The long distance between supply chain partners and slow modes of transportation induce not only high transportation costs and in-transit inventory, but also high inventory carrying costs in terms of safety stocks at the final customer location.

The great distances between supply chain partners present a high variability of transportation times that can hurt suppliers in terms of service levels and final customers in terms of safety stock costs. Moreover, the transportation process is carried out either by ships, trucks, pipelines, or railroads. In many instances, a shipment has to exploit multiple transportation modes before reaching the final customer's location. "Very few industries 92 deal with that kind of complexity in shipping," said Doug Houseman, a senior manager at the consulting firm Accenture (Morton, 2003) Such constraints on transportation modes in this type of industry induce long lead times from the shipping point to the final customers' location compared to other industries. Hence, considering the amount of inflexibility involved, meeting the broadening prospect of oil demand and its derivate while maintaining high service-levels and efficiency is a major challenge in the petroleum industry.

2.1 Optimization of the supply chain of service stations

The hydrocarbon supply chain is vertically integrated and covers activities ranging from exploration to processing in refineries and product distribution with an extensive logistics network. The entire supply chain is divided into forward and backward operations. In view of the complexity and interactions of the hydrocarbon supply chain, a sub optimality of one of the chain's component links can impact the whole chain's activities. Forward activities include exploration, crude oil production, and refining.

Our work will focus on this part (backward activities) include fuel storage in warehouses, transportation, and distribution of different types of fuels to the final customer.

- **Storage:**

The storage of fuel is a mean of collecting the product before distributing it to service stations, so it is stored in depot centers, after being transformed and processed in refinery centers.

- **Transportation:**

This is the transportation of the fuel from the storage centers to the service stations, done with tanker trucks, which supply their tanks from the orders received.

- **Distribution to service stations:**

The distribution of fuel represents the stage of replenishment of service stations, which consists of delivering them taking into account their requirements and their possible constraints in terms of time and quantity. Therefore, the objective of the distribution companies is to satisfy all orders, in terms of quantities and times, in other words, to deliver the requested quantities in the shortest possible time. This challenge implies a reactive Supply Chain that reacts quickly to demand and is able to quickly adapt the volumes and variety of products to the requirements of the service stations, while taking into account their possible constraints. The distributor will focus on optimizing the distribution of fuel to the service stations in order to be able to fulfill orders within a specific time frame. The optimization of the distribution therefore requires a strict organization and considerable resources (planning of routes, assignment of trucks to service stations and optimal distribution quantities to be delivered).

The second challenge for fuel distributors is to meet the demand of service stations at any time, while controlling their stocks (eliminating stock-outs), and therefore increasing their service quality, which requires a reactive supply chain.

Finally, more efficient and effective supply chain practices in the oil industry are important factors in ensuring a continuous supply of crude oil, reducing lead times and lowering production and distribution costs. Due to the inflexibility of the oil industry's supply chain network, logistics is a major challenge. However, it is only one of many challenging factors. Also relevant are the integrated process management, information systems and information sharing, organizational re-engineering and organizational reorientation.

Combinatorial optimization

The combinatorial optimization is a subfield of mathematical optimization that consists of finding an optimal object from a finite set of objects, where some of feasible solutions is discrete or can be reduced to a discrete set. It is an emerging field at the forefront of combinatorics and theoretical computer science that aims to use combinatorial techniques to solve discrete optimization problems. A discrete optimization problem seeks to determine the best possible solution from a finite set of possibilities.

Within the distribution network is composed of a set of interconnected actors and activities, whose mission is the physical transfer of finished products from the manufacturer to its customers. The objective is to ensure that the product desired by the customer is in the right place, at the right time, with the right quantity and at the best cost. The distribution network is characterized by information flows that drive and control these physical operations, such as demand forecasts, planning operations or even the administrative processing of orders and inventory management. In order to optimize a distribution network, it must adopt a route planning approach that takes into account certain constraints related to transportation time and cost, distances covered and delivery times, with the aim of increasing the customer satisfaction rate.

3 Fundamental concepts of predictive analytics

We apply the time series analysis technique in companies for demand forecasting when historical data for a product or product line is available and trends are evident to help organizations to understand the underlying causes of trends or patterns over time. “Seasonal variations in demand, cyclical patterns, and key sales trends can all be identified using time series analysis.”
Predictive analytics is a type of technology that makes future predictions about unknowns. Artificial intelligence (AI), data mining, machine learning, modelling, and statistics are among the tools used to make these predictions.

Demand forecasting is a type of predictive analytics technique that aims to analyses and predict future customer demand over a set period of time based on previous data and other data in order for corporate supply chain and business management to make better supply decisions. There are two types of demand forecasting methods: qualitative and quantitative methods.

Underfitting is a scenario in data science where a data model is unable to capture the relationship between the input and output variables accurately, generating a high error rate on both the training set and unseen data. (CloudIBM)
Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose. (CloudIBM)

3.1 Predictive Models

Predictive modeling is a process of mathematics used to predict future events or outcomes by analyzing the patterns in a given set of data inputs. (Satvik Garg,Himanshu Jindal ,Time Series Forecasting Models , 2021)

Sequential Model

Sequence learning is the study of machine learning algorithms designed for sequential data.

I. Recurrent Neural Networks

Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist. At the bottom, a chunk of neural network A, looks at some input x_t and outputs a value h_t

A loop allows information to be passed from one step of the network to the next.

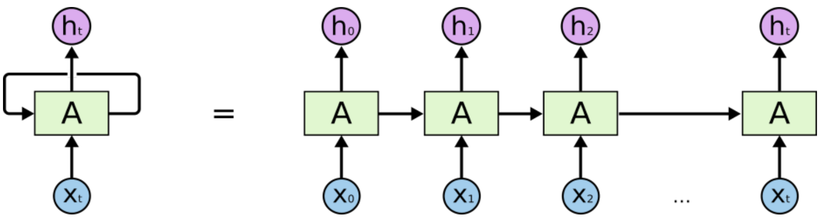


Figure 3 Recurrent neural network Architecture

Also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states[2]. They are typically as follows:

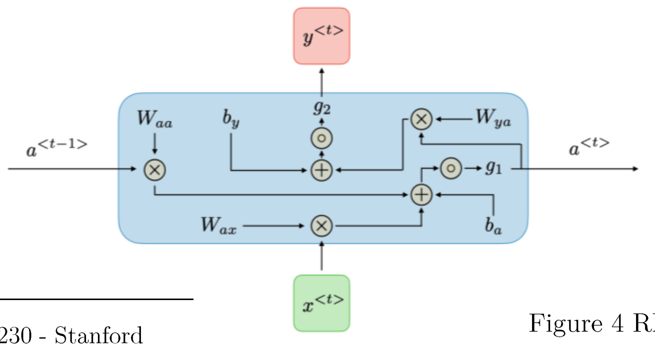


Figure 4 RNNs cell

[2] Figure Source: CS 230 - Stanford

For each time step t, the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$a^{<t>} = g_1(W_{aa} \quad a^{<t-1>} + W_{ax}\ x^{<t>} + b_a)$ And $y^{<t>} = g_2(W_{ya}\ a^{<t>} + b_y)$

Where W_{ax},W_{aa},W_{ya}, b_a, b_y are coefficients that are shared temporally and g_1, g_2 activation functions.

With Loss function of all time steps defined based on the loss at every time step equal to:

$$\boldsymbol{L}(\tilde{y}, \boldsymbol{y}) = \sum_{1}^{T_y} \boldsymbol{L}(\tilde{y}^{<t>}, \boldsymbol{y}^{<t>})$$

The most common activation functions used in RNN modules are described below:

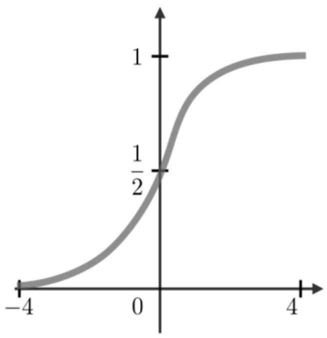
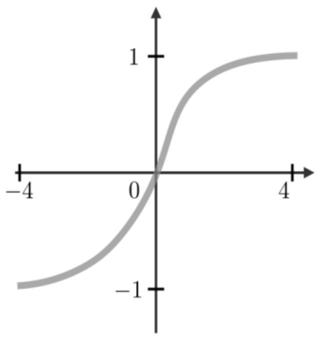
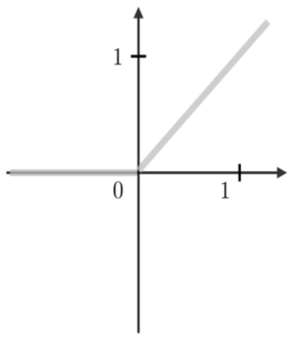
Sigmoid	Tanh	RELU
$g\ z = \dfrac{1}{1+\ e^{-x}}$	$g\ z = \dfrac{e^{x} -\ e^{-x}}{e^{x} +\ e^{-x}}$	$g(z) = \max(0, z)$
		

Table 2 activation functions table

Table 3 The positive and cons of RNN architecture:

Advantages	Drawbacks
*Possibility of processing input of any length *Model size not increasing with size of input *Computation considers historical information and weights are shared across time	*Computation being slow *Difficulty of accessing information from a long time ago *Cannot consider any future input for the current state

RNNs, in theory, can completely deal with such "long-term dependencies." Sadly, RNNs do not seem to learn them in practice. (Hochreiter[German], 1991)and (Bengio, 1994) studied the problem in depth and identified some pretty fundamental reasons it might be difficult, in our context, the vanishing and exploding gradient phenomena are frequently encountered. Because of the multiplicative gradient, which can be exponentially decreasing/increasing with respect to the number of layers, it's difficult to capture long-term correlations. Thankfully, LSTMs don't have this problem !

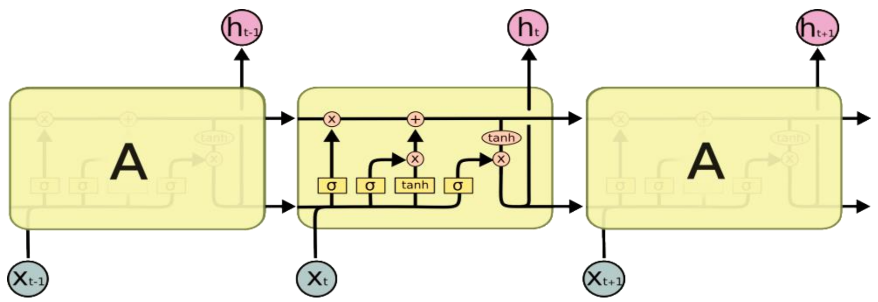
II. LSTM Networks

The Long-Short-Term Memory (LSTM) is a special kind of RNN model with two memory units that allow it to retain information over long periods of time. You can think of the LSTM as an "artificial hippocampus." They were introduced by (Hochreiter & Schmidhuber, 1997). LSTMs are explicitly designed to avoid the long-term dependency problem "Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!"

How can an artificial LSTM network improve the efficiency of our business?

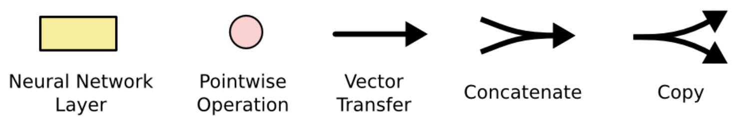
An artificial LSTM network can be used in many different ways in the business world. It can be used in many areas such as speech recognition, language translation, recommendation engines, and computer vision. It can also be used in the design of self-driving cars

Figure 5 Standard LSTM structure [3]



The repeating module in an LSTM contains four interacting layers.

LSTM notation:

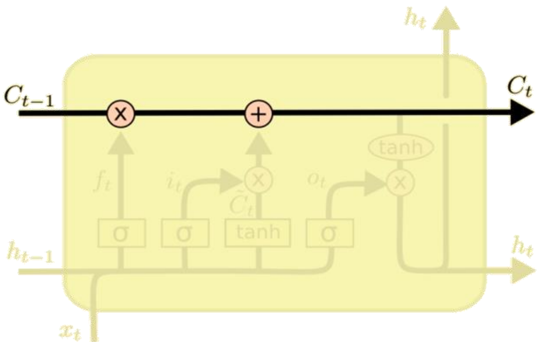


In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines

[3] Figure Source: CS 230 - Stanford

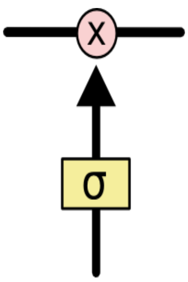
merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

[4]**The Core Idea behind LSTMs:** is the cell state, the horizontal line across the top of the diagram. It's a little like a conveyor belt. It runs along the entire chain, with only minor linear interactions. It is very easy for information to flow along this line



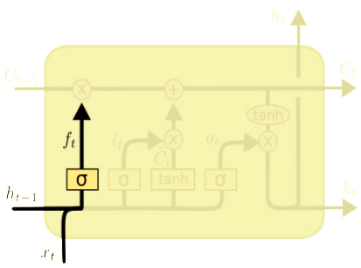
without being modified.

The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of:



The sigmoid layer produces numbers between zero and one, which describe how many of each component to pass. A value of zero means "pass nothing", while a value of one means "pass everything!" and to decide what information we will discard from the cell state. This decision is made by the "forget gate layer". It looks at **xt**, and outputs a number between **1** for each number in the cell state: **1** represents "keep this completely" while a 0 represents "get out of this completely".

Figure 6 Cell horizontal line [1]

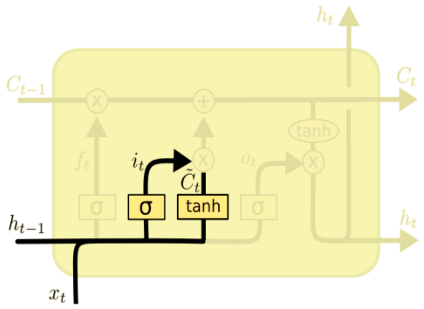


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Let's return to our example of a sequence model that tries to predict the next value based on all previous values.

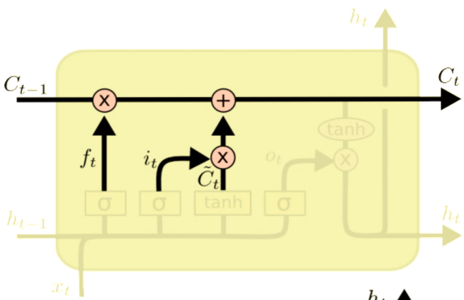
[4] Figure Source: CS 230 - Stanford

The following step is to decide what new data we will store in the cell state. This step has two parts. First, a sigmoid layer called the "input gate layer" chooses which values to update. Second, a tanh layer creates a vector of new candidate values, C~t, that could be added to the state. [5]

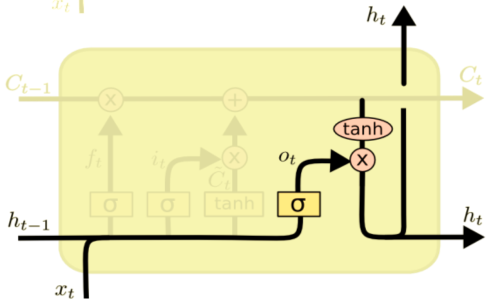


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Now it's time to upgrade the old cell state, Ct-1, to the new cell state Ct. The previous steps have already decided what to do, we just need to do it really. We multiply the old state by ft. (forgetting) the things we decided to forget earlier. Then we add it to Ct. In the case of the sequential model, this is where we drop the old information and add the new information, as we decided in the previous steps. [6]



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Finally, we have to decide what we are going to generate. This output will be based on our cell state, but will be a Filtered output. First, we run a sigmoid layer that decides which parts of the cell state we are going to output. [7]

[5] Figure Source: CS 230 - Stanford
[6] Figure Source: CS 230 - Stanford
[7] Figure Source: CS 230 - Stanford

Then, we run the cell state through tanh (to push the values to be between and multiply them by the output of the sigmoid gate, so that we only output the parts we decided.

Demand forecast accuracy

The forecast error needs to be calculated using actual sales as a base. There are several forms of forecast error calculation methods used, namely Mean Percent Error, Root Mean Squared Error, Tracking Signal and Forecast Bias.

RMSE The root-mean-square deviation or root-mean-square error is a frequently used measure of the differences between values predicted by a model or an estimator and the values observed.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

MAE In statistics mean absolute error is a measure of errors between paired observations expressing the same phenomenon.

$$MAE = \frac{1}{n} \sum_{j=1}^{n} |y_j - \hat{y}_j|$$

MAPE The mean absolute percentage error, also known as mean absolute percentage deviation, is a measure of prediction accuracy of a forecasting method in statistics

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

RRSE The Root Relative Squared Error is defined as the square root of the sum of squared errors of a predictive model normalized by the sum of squared errors of a simple model. In other words, the square root of the Relative Squared Error (RSE). , where: n: represents the number of observations

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

3.2 Neural combinatorial optimization

The combinatorial optimization holds an important place in operational research, in discrete mathematics and in computer science. Indeed, combinatorial optimization problems represent a class of very difficult problems to solve and many practical problems can be formulated in the form of a combinatorial problem.

A combinatorial optimization problem, in a discrete space of feasible solutions, consists in finding the best solution (or a set of best solutions).The main specification of an optimization problem is the minimization (maximization) of an objective function. The objective function is the function to be minimized with constraints can be defined as follows: Min/Max f(x), where $x \in X$,

Markov Decision Process

MDP is a discrete-time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker (Wrobel, 1984). MDPs are useful for studying optimization problems solved via dynamic programming (Bellman, 1957).

Terminology

First things first, before even starting with MDPs, we'll quickly glance through the terminology that will be used throughout this article:

- **Agent:** An RL agent is the entity which we are training to make correct decisions.
- **Environment:** The environment is the surrounding with which the agent interacts. The agent cannot manipulate the environment; it can only control its own actions.
- **State:** The state defines the current situation of the agent.
- **Action:** The choice that the agent makes at the current time step. We know the set of actions (decisions) that the agent can perform in advance.
- **Policy:** A policy is the thought process behind picking an action. In practice, it is a probability distribution assigned to the set of actions. Highly rewarding actions will have a high probability and vice versa.

Note that if an action has a low probability, it doesn't mean it won't be picked at all. Just that it is less likely to be picked.

3.3 Transformers for decision making

The most competitive models of neural sequence transduction have an encoder-decoder structure. Here, the encoder maps an entry sequence of symbols representations $(x1, \ldots, xn)$ to a sequence of continuous representations $z = (z1, \ldots, zn)$. Given z, the decoder then generates a sequence output (y1, ..., ym) of symbols one element at a time.

Transformers in vrp aim to solve sequence-to-sequence problems while easily dealing with long-range dependencies, and are also increasingly becoming popular for tasks such as reinforcement learning. RL estimates single steps using the Markov properties to work on a task over time.

While the Transformer architectures can effectively model sequential data, their self-attention mechanism enables the layer to assign a reward by maximizing the dot product of the query and key vectors and forming state-return combinations. Therefore, Transformers can operate efficiently with a distracting reward. Studies have found that Transformers provide better generalization and transfer capabilities because of their ability to model a wide distribution of behaviors.

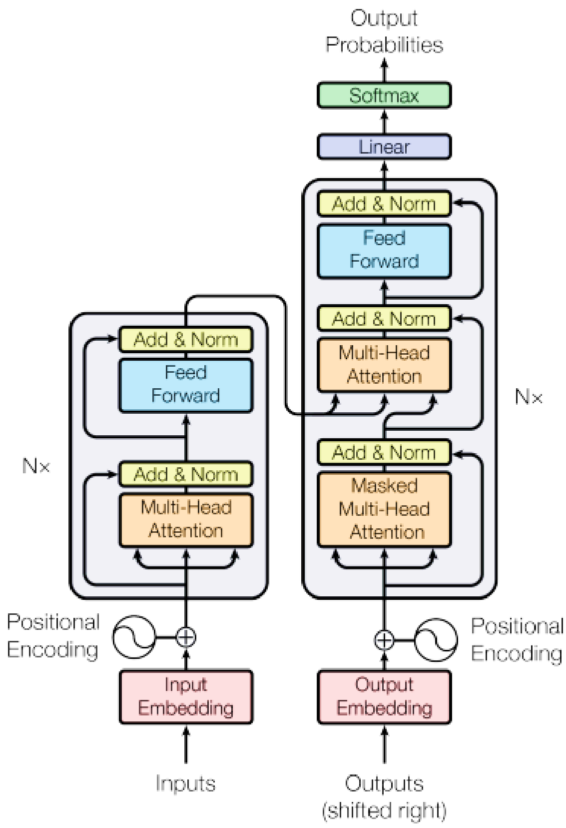


Figure 7 Transformers architecture 7

The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 7, respectively.

The encoder is composed of a stack of N identical layers. Each layer has two sublayers. The first one is a multi-headed self-attention mechanism, and the second one is a simple, fully position-connected, feed-forward network. We employ a residual connection around each of the two sub-layers, followed by layer normalization. That is, the output of each sub-layer is LayerNorm(x + Sublayer(x)), where Sublayer(x) is the function performed by the sublayer itself. To facilitate these residual connections, all the sublayers of the model, as well as the integrating layers, produce outputs of dimension d_{model} .

The decoder is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. We also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking combined with fact that the output embedding's are offset by one position, ensures that the predictions for position **i** can depend only on the known outputs at positions less than i. (Attention Is All You Need)

Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. (Attention Is All You Need)[8]

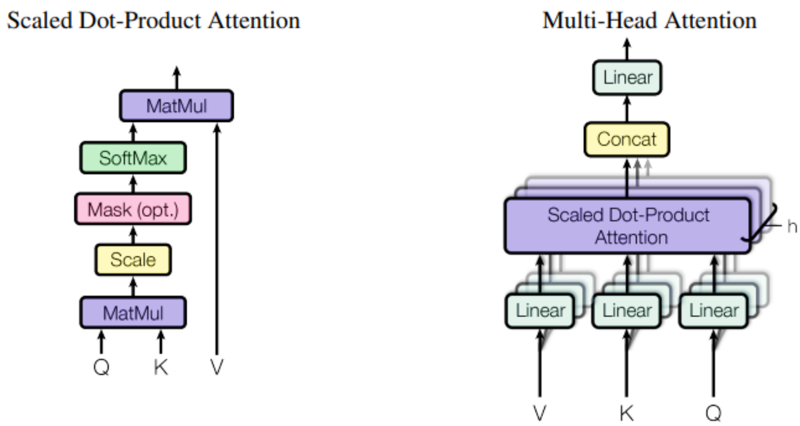


Figure 8 Attention layer

In (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 8). The input consists of queries and keys of dimension dk, and values of dimension dv. We compute the dot products of the query with all keys, divide each by √dk, and apply a softmax function to obtain the weights on the values. In practice, we compute the

[8] Ashish Vaswan and Al., "Attention Is All You Need", Google Brain, 2017, page 04

attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V.

We compute the matrix of outputs as:

$$\mathrm{Attention}(\mathrm{Q},\mathrm{K},\mathrm{V}) \; = \; \mathrm{softmax}(\frac{\mathrm{QKT}}{\sqrt{\mathrm{dk}}})\mathrm{V} \; (1)$$

The two most commonly used attention functions are additive attention, and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\sqrt{\frac{1}{\mathrm{dk}}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer.

While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code. While for small values of dk the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of dk.

We suspect that for large values of dk, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients. To counteract this effect, we scale the dot products by $\sqrt{\frac{1}{\mathrm{dk}}}$.

Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to dk, dk and dv dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding dv-dimensional output values. These are concatenated and once again projected, resulting in the final values, as depicted in Figure 8.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. (Attention Is All You Need)

Decision making transformers Steps :

1) Each modality (return, state, or action) is passed into an embedding network (convolutional encoder for images, linear layer for continuous states).

2) Embeddings are processed by an autoregressive transformer model, trained to predict the next action given the previous tokens using a linear output layer.

Instead of training policy in a RL way, authors aim at sequence modeling objective, with use of sequence modeling algorithm, based on Transformer (the Transformer is not crucial here and could be replaced by any other autoregressive sequence modeling algorithm such as LSTM)

- The algorithm is looking for actions conditioning based on the future (autoregressive) desired reward.
- Presented algorithm requires more work/memorizing/learn by the network to get the action for a given reward (two inputs: state and reward), comparing to classical RL where the action is the output based on the maximized reward for a policy. One input: state, to train the policy. However, authors tested how transformers can cope with this approach minding recent advancements in sequence modeling.
- Decision Transformer is based on **offline RL** to learn from historical sequence of (reward, state, action) tuples to output action, based on imitation of similar past reward and state inputs and action outputs. Solution is learning about sequence evolution in a training dataset by learning from agents' past behaviors for similar inputs: state and the presently required reward, to output an action.

Conclusion

The objective of this chapter was to introduce the different notions and tools related to the predictive analytics, in order to improve the performance of a service station and controlling the demand by a reliable forecast using deep learning tools, conditioned by the availability of information in real time, and this, by using the adapted IoT technologies. Moreover, in order to achieve an optimization of the Supply Chain, it is necessary to ensure a good coordination between the distributor and the service stations. Hence the importance of optimizing fuel distribution, and this, by using the tools of operational research, namely mathematical models and using the neural combinatorial optimization by developing mathematical heuristics using reinforcement learning.

Chapter 03: Improvement of the responsiveness of stations and the network distribution

1 Introduction

The problems facing companies always come in the form of data, the constraints that must be taken into account and an objective to be achieved. Modeling is therefore a translation of the parameters of the problem into a language that is accessible by the solving method used, or else, it is a way of writing the problem in a form that introduces its resolution for that we begin by introducing the way of collecting and forming the data.

However, in practice, perfect conditions never exist, since the problems by their specific characteristics, must satisfy a very large number of constraints, which cannot in any case be all taken into account, in which case the resulting model will not concretely reflect the problem. Finally, the modeling of a problem must be able to give an interpretation to the obtained solutions in terms of concrete solutions.

2 Stations improvement using IOT tools

The Internet of Things (IoT)

Can be defined as “a set of interconnected physical devices that can be used to monitor, report, send and exchange data”. IoT devices are typically connected to computers via data network or Wi-Fi. In a supply chain, IoT devices are an effective way to track and authenticate products. They can also monitor product storage conditions, improving quality management throughout the global distribution channel. As a result, it is much easier to understand where goods are, how they are stored and when they can be expected at a specific location. [9]

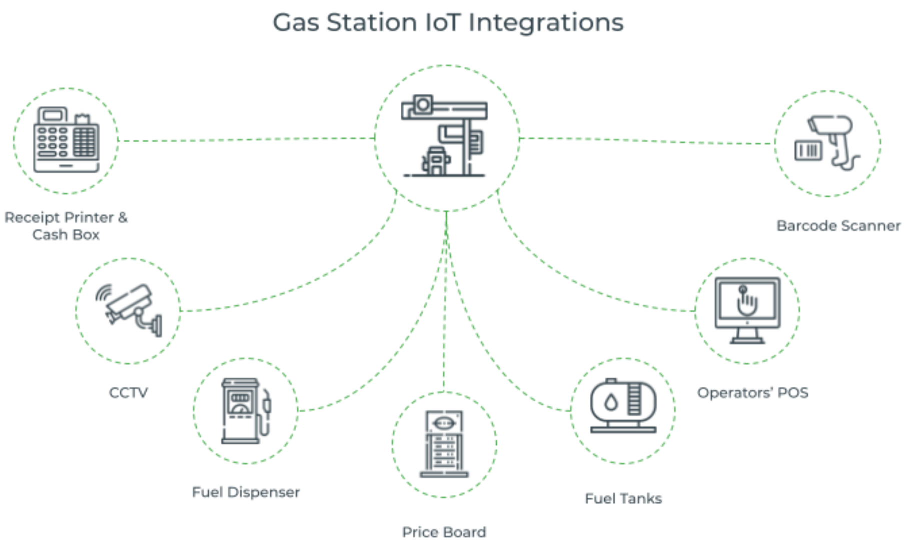


Figure 9 Station IoT integration

[9] Figure source: Jevera, “retail-gas-station-as-a-profitable-business-model”, P01

In order to ensure better coordination between the distributor and the service stations, the information flow becomes the essential element of synchronization.
In fact, the integration of information will allow service stations and their distributor to make the various existing exchanges more reliable, in other words, information relating to the quantities ordered, the types of products requested and the deadlines to be met. It is necessary to control this information before acting on the stocks and on the distribution.

This control will imply a better planning of the demand. As for the service stations' objective, it is on the one hand to manage to synchronize the flows coming from the distributor and those coming from the customers, and to act quickly on the demand, by ensuring the availability of the product at each moment, in order to increase the satisfaction rate of the consumers on the other hand.

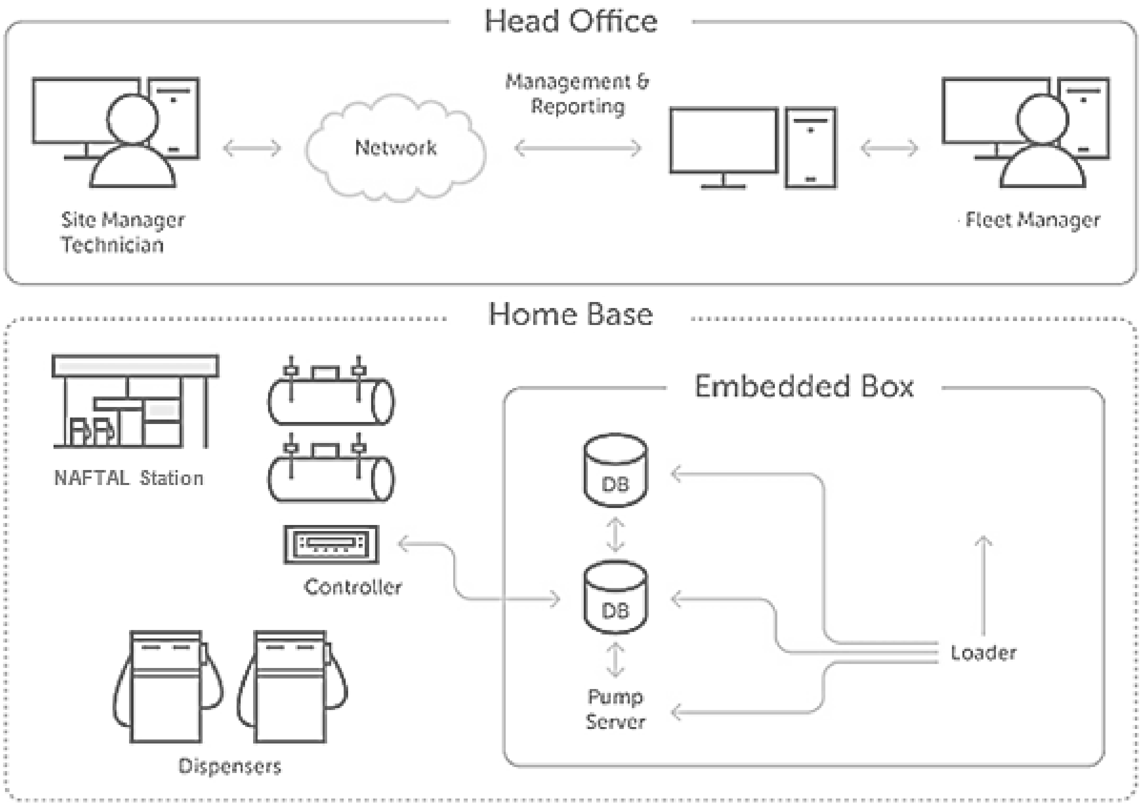


Figure 10 Station digitalisation network layout

To do this, service stations must build up a stock that allows them to anticipate risks related to over-stocking and stock-outs, and this, by using a safety stock. The integration of these risks, will allow ensuring at best the availability of the product, which will ensure in its turn a service rate, thus a better reactivity!

[10] Figure source Web : Kaspersky, « Station Network layout"

To solve this problem, one must first start by interpreting all these parameters like (barcode scanner, tanks sensors …), trying to transform them into manageable forms. Therefore, the first step in solving a problem is to project it into a workable space, which is called the modeling associated with the problem by developing a database, where our data is stored and visualize them in the dashboards.

We were interested in collecting, processing and analyzing all the information related to the main components of the problem, i.e., the fleet of trucks, the customers, the routes and the depots. In the following, we present the collected information so that it can be used by our mathematical model.

IoT devices allow:

- To track the speed of movement and receipt of goods ;
- Administer goods immediately upon receipt;
- Simplified planning of supply and demand, (stakeholders know when they can expect to receive (stakeholders know when they can expect to receive and process goods);
- Improved quality management by maintaining raw materials and processed products in optimal way.

To conclude, even if the implementation of data collecting is wrong and using very robust methods for solving a given problem is very important, data collection and preparation is still the first step on the way to solving it in an right way.

2.1 Data security

Anomaly detection is the determination of whether something has deviated from the "norm". Anomaly detection using neural networks is modeled in an unsupervised tools (B. Lindemann , B. Maschler, N. Sahlab), as opposed to supervised learning, where there is a one-to-one connection between the input feature samples and the corresponding output labels. The basic assumption is that normal behavior, and thus the amount of "normal" data available is the norm and that anomalies are the exception to the norm to the point that modeling "normality" is possible.

We will use an autoencoder deep learning neural network model to identify vibrational anomalies from sensor readings. The goal is to predict future bearing failures before they occur. Autoencoder's are an unsupervised learning approach, although they are trained using supervised learning methods. The goal is to minimize the error of reconstruction as a function of a loss function, such as mean square error.

In this subsection, we will try to detect anomalies in the fuel demand historical time series data with an LSTM autoencoder.

Auto-encoder Anomaly detection using LSTM

The steps we will follow to detect anomalies in historical demand of Garidi station data using an LSTM auto-encoder:

- First, we split data to 70% for training and 30% for testing data.
- Train an LSTM autoencoder on the historical demand data from 2021-01_02 to 2021–03–01. We assume that there were no anomalies and they were normal.
- Using the LSTM autoencoder to reconstruct the error on the test data from 2021–03–01 to 2021–03–31.
- If the test data reconstruction errors exceed the threshold, we refer to the data point as an anomaly.

We will break down an LSTM autoencoder network[11] to understand them layer-by-layer.

```
From tensorflow import keras
from sklearn.preprocessing import StandardScaler
import plotly.graph_objects as go

np.random.seed(1)
tf.random.set_seed(1)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout, RepeatVector, TimeDistributed

df = pd.read_csv('data/Garidi.csv')
df = df[['Date', 'Close']]
df['Date'] = pd.to_datetime(df['Date'])
df['Date'].min(), df['Date'].max()
```

To understand more the demand we need to **make the time series visual**

```
fig =go.Figure()

fig.add_trace(go.Scatter(x=df['Date'], y=df['Demand'], name='Demand station'))
fig.update_layout(showlegend=True, title='Station demand 2021')
fig.show()
```

Then, we split the data to train and test

```
train, test =df.loc[df['Date'] <= '2021-09-03'], df.loc[df['Date'] > '2021-09-03']
```

[11] Source: susanli2016

- We standardize the data using StandardScaler() function.
- Create sequences by converting the input data into 3-D array with *TIME_STEPS*. The shape of the array i.e. [samples, TIME_STEPS, features] , for LSTM network as required.
- Our Autoencoder LSTM expects input sequences with N-time steps and one feature and outputs a sequence with N-time steps and one feature.
- The RepeatVector() function is repeat the inputs N-times.

```
model = Sequential()

model.add(LSTM(128, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(rate=0.2))
model.add(RepeatVector(X_train.shape[1]))
model.add(LSTM(128, return_sequences=True))
model.add(Dropout(rate=0.2))
model.add(TimeDistributed(Dense(X_train.shape[2])))
model.compile(optimizer='adam', loss='mae')
```

- Then we train our model using train historical demand for the both products and plotting them.

Our approach to **Determine Anomalies** is as follows:

- Find MAE loss on the training data.
- Make the max MAE loss value in the training data as the reconstruction error threshold.
- If the reconstruction loss for a data point in the test set is greater than this reconstruction error threshold value then we will label this data point as an anomaly.

```
X_train_pred = model.predict(X_train, verbose=0)
train_mae_loss = np.mean(np.abs(X_train_pred - X_train), axis=1)

plt.hist(train_mae_loss, bins=50)
plt.xlabel('Train MAE loss')
plt.ylabel('Number of Samples');
threshold = np.max(train_mae_loss)
print(f'Reconstruction error threshold: {threshold}')
```

To know the number of the anomaly point:

```
anomalies = test_score_df.loc[test_score_df['anomaly'] == True]
anomalies.shape
```

3 Improvement of station activities

In the context of IoT enabled service stations, producing a huge set of historical data available for analysis of the demand pattern. The demand here will be the volume of stock in the stations. For having an accurate idea of demand that helps in planning of the service chain process, where demand can change frequently and periodically, a clear demand is known well in advance are dependent on the historical pattern of demand and a set of other correlated factors which influence the resulting demand.

These historical data for the volume of stock in the service stations is a multivariate time series data continuous on a daily basis. The readings from different sensors constitute the data collected for different variables. Any change in demand in a bay can have an effect on the demand in other bays and having advance visibility of the demand can help with managing the resources, which in turn will enhance customer experience.

We will develop a diagnosis in order to detect any possible improvements to the service station then present and formulate the model that will be used for evaluating the forecasting techniques based on the analysis of the historical data, describes the process of experimenting to get to the best forecasting technique for our case and documents the MAPE results of each technique.

Stations demand forecast

The losses caused by stock-outs are mainly related to the variability of delivery times caused by the distributor, as well as by the forecasting error made by the service station. To solve the problems related to stock-outs, we propose to set a security stock taking into account on the other hand the uncertainty of the real demand and of the replenishment delay and the need to satisfy the demand.

To do this, we must first know the nature of the demand, i.e., forecast the future daily demand. And this, based on a history provided by the station in particular. In addition to this forecast, it is necessary to determine the law regulating delivery times, using a history containing the difference between the time of placing the order and the time of its reception. Once these data are validated, we will be able to determine the appropriate safety stock.

In the context of IoT enabled service stations, various forecasting techniques exist and are used for predicting from a huge set of historical data available for analysis of the demand pattern. For this reason they include classical methods, e.g. ARIMA and Fb prophet, artificial intelligence based on recurrent neural networks e,g, LSTM and transformers using methods to calculate the error that exists in the predicted results like mean absolute error, mean absolute percentage error and mean squared error. With munching the effect of parameters that depend on their correlation with the stock data that may be a factor in the correct prediction of the demand for that we estimate these parameters using the Genetic algorithm.

In this section, our goal is to build a stock prediction service station model where the network and infrastructure based on IOT technologies is being maintained by one of our partners. The model will be incorporated into a prediction framework together with an analytical dashboard to provide additional insight from the large volume of data being collected from these IoT infrastructures.

We will build together a several models of prediction based on the popular prediction algorithms based on work of Garg, S., & Jindal, H (Evaluation of time series forecasting models for estimation of pm2. 5 levels in air, 2021), (Forecasting at scale, 2017), (Preston, 1991) and (Brownstone, David S. Bunch, and Al., 1994) with modification in codes using Different model of evaluation like metric, calculating the mean absolute percentage error (MAPE) of the demand prediction produced by these various forecasting models considering the type of problem, Since forecasting requires regression models, accuracy is the measure that is normally used. We have decided to use MAPE for this purpose, because of its intuitive interpretation of relative error, converting the data from a multivariate time series data continuous on a daily basis to univariate data by collecting only the count of the number of stock per station per hour and day. We take the MAPE results from the ARIMA and Recurrent Neural Networks.

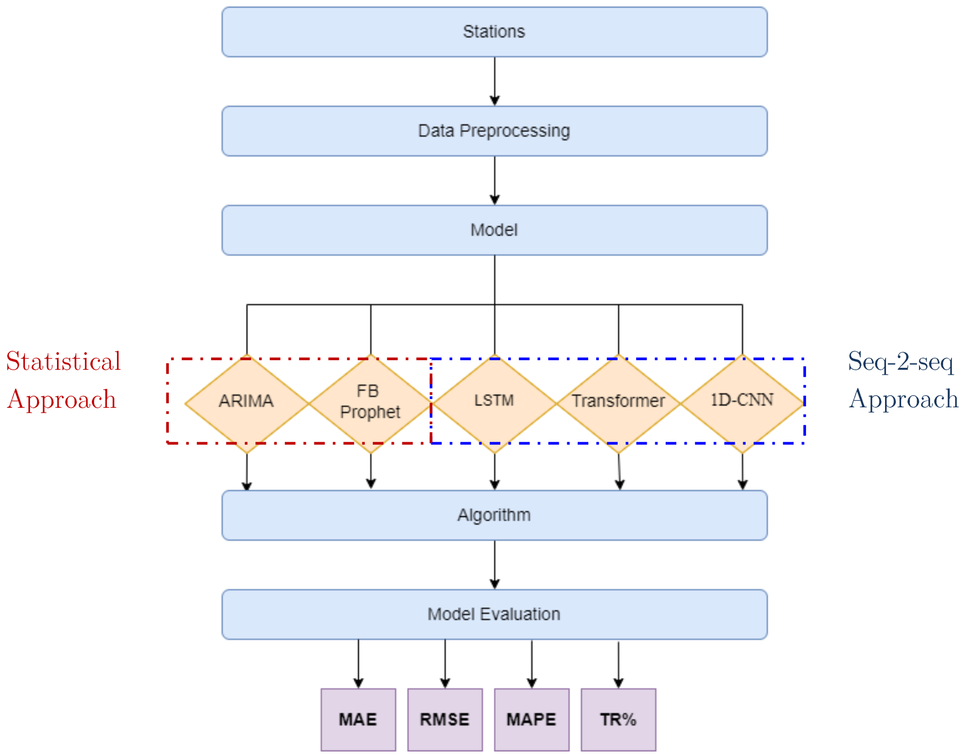


Figure 11 Flowchart of the framework adopted

Our goal is to evaluate each algorithm and find the best model for our problem. For that a large set of experiments were performed with different parameter settings for each of the algorithms and the best results were recorded together with the parameter settings that found them. In this section, we detail the logic used for each algorithm to find the best configuration. We also present the best results found by each algorithm. The python based scikit-learn framework is being used for experiments.

I. Statistical approach

Autoregressive Integrated Moving Average

ARIMA is a statistical technique analysis model that uses time series data to better understand the data set to anticipate the future by predicting outcomes using previous data. It is based on the statistical idea of serial correlation, which states that previous data points have an impact on future data points using different parameters (Auto Arima), as:

p: the number of lag observations in the model; also known as the lag order.

d: the number of times that the raw observations are different; also known as the degree of differencing.

q: the size of the moving average window; also known as the order of the moving average.[12]

Algorithm: Pseudo code ARIMA

Input: Data, Features
Output: Evaluation metrics of the predicted data
Data split : 75% train and 25% test data
Train
1: count ← length(Data) ∗ 0.75
2: X ← Data(0 : count)
3: Z ← F eatures(0 : count) Test
4: x ← Data(count :)
5: z ← F eatures(count :)
6Model fitting
7: model ← auto arima(X, exogenous=Z)
8: model.f it() 16: forecast ← model.predict(n periods = len(x), z)
9 end
10: return rmse, mae, mape, rrse

[12] Pseudo-code Sources: Satvik Garg , Himanshu Jindal, Evaluation of Time Series Forecasting Models P05

Table 4 ARIMA Evaluating forecast accuracy

	MAE	RMSE	MAPE	TR%
Product 01	1025.61	1390.41	0.08	91.89
Product 02	3017.43	3716.496	0.096	90.40

FB Prophet

The general idea of the model is similar to a generalized additive model (Forecasting at Scale: How and Why We Developed Prophet for Forecasting at Facebook). The "Prophet Equation" fits, as mentioned above, trend, seasonality and holidays. This is given by (The Prophet on Walmart — Comprehensive Intro to FbProphet., 9 July 2019), then y(t) = g(t) + s(t) + h(t) + e(t); y(t) is the forecast, where:

- g(t) refers to trend (changes over a long period of time)
- s(t) refers to seasonality (periodic or short term changes)
- h(t) refers to effects of holidays to the forecast
- e(t) refers to the unconditional changes that is specific to a business or a person or a circumstance. It is also called the error term.

Algorithm: FB Prophet Pseudocode

Input: Data, Features
Output: Evaluation metrics of the predicted data
Data split : 75% train and 25% test data
Train
1: count ← length(Data) ∗ 0.75
2: X ← Data(0 : count)
3: Z ← F eatures(0 : count) Test
4: x ← Data(count :)
5: z ← F eatures(count :)
Model fitting
6: parameters ← Hyperparameters
7: for each p in parameters do
8: model ← P rophet(p, interval width = 0.95)
9: model.add regressor(Z)
10: model.f it(X)
11: forecast ← model.predict(z)
12: return rmse, mae, mape, rrse
13: **end**

Table 5 Evaluating forecast accuracy for Station - FB Prophet

	MAE	RMSE	MAPE	TR%
Product 01	759.89	982.53	0.091	90.918
Product 02	2279.446	3041.55	0.117	88.279

II. Sequence two sequence approaches:

LSTM

A demand forecasting method based on multi-layer LSTM networks is proposed. The proposed method improves the forecasting accuracy. It has a strong ability to capture nonlinear patterns in time series data. The empirical results show that the method outperforms other standard techniques.

Algorithm: Pseudocode LSTM

1: input: normalized data, Split training and validation subsets.
2: output: best solution
3: begin
4: Initialize
5: Initialize the parameters
6: Parameters:
For ts in list [time steps]: For ihl in list [memory cells i in layer hl]:
For j in list [dropout values]:
For k in list [activ. functions in output neuron]:
For l in list [optimizers]:
For m in list [error functions]:
For n in list [epochs]:
For o in list [batch sizes]:
For p in range [repetitions]:
7: Build training (ts) and validation (ts) lists.
8: Compile LSTM network (hl, ihl, j, k, l, m, n, o, p).
9: Train the LSTM network dynamically ....
10: Predict to validation (ts) and training (ts)....
11: Denormalize forecasts.
12: Calculate error measures ...
13: Append results to previously initialized list.
14: Save results summary in .csv file.
15: end
16: output best parameter set for LSTM model
17: end

Table 6 Evaluating forecast accuracy for Station – LSTM

	MAE	RMSE	MAPE	TR%
Product 01	1131.005	1419.821	0.060	93.96
Product 02	3657.127	4577.594	0.069	93.03

Transformers

A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV). Among multiple advantages of transformers, the ability to capture long-range dependencies and interactions is especially attractive for time series modeling, leading to exciting progress in various time series applications (Neo Wu, B. Green, 2020).

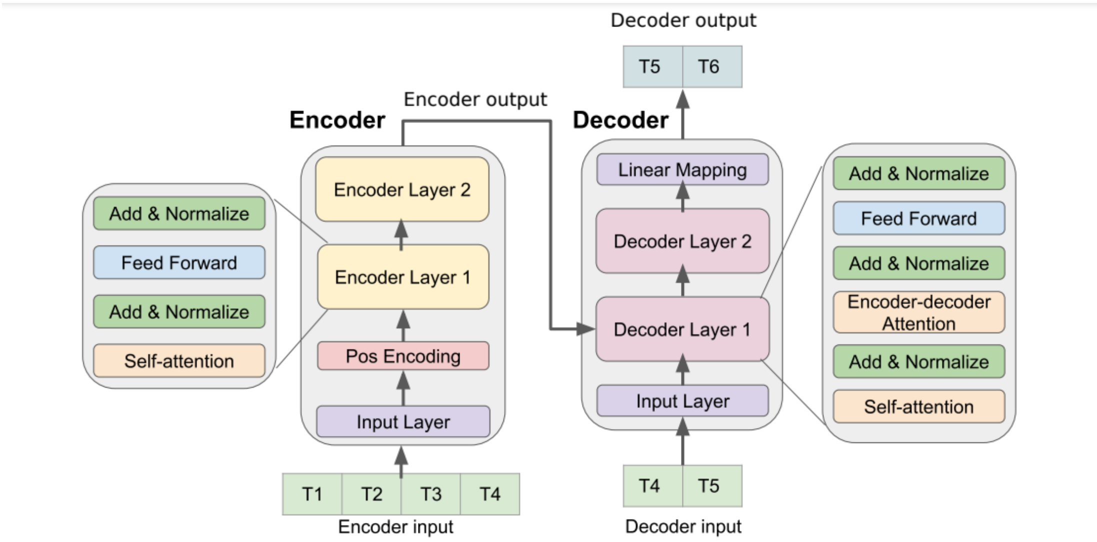


Figure 12 Architecture of Transformer based forecasting model.

[13]

Table 7 Transformers Evaluating forecast accuracy for Station

	MAE	RMSE	MAPE	TR%
Product 01	1531.605	1269.425	0.070	92.96
Product 02	3437.15	4247.574	0.087	91.3

[13] Deep transformer models for time series forecasting: arXiv preprint arXiv:2001.08317 p04

1D- Deep Convolutional Networks (1D-CNN)

Any image available in numerical form is in fact a matrix of pixel values. Each pixel value can go from 0 to 255 depending on the intensity of the pixel. Each picture also includes channels that depend on the color composition of the image. A gray image has a channel since each channel corresponds to the colors it contains. A color image has three channels including red, blue and green colors. A Convolutional Neural Network understands each image as a matrix of pixel values in the dimension of the width, length and number of channels of the image.

So if we can convert the 1D time series sequence into an input image matrix form, we would be able to apply a CNN model for the predicting problem. We will now be discussing the methodology and a sample test sequence on which we will apply our model...

Algorithm: Pseudocode 1D-CNN

1: Import Keras libraries
2: input: normalized data, Split training and validation subsets.
3: output: best solution
4: begin
5: Define a function that **extracts features** and **outputs from the sequence**.
6: **Reshape** the inputX in a format that is acceptable to CNN models
7: Design the CNN model architecture of convolutional layers(Conv-1D);
pooling(max-pooling in our case);
flattening layer;
and the fully connected neural layers.
8: model ← Sequential()
9: model.add(Conv1D(f ilters = 128, kernel size = 2, activation = ’relu’))
10: model.add(MaxP ooling1D(pool size = 2))
11: model.add(F latten())
12: model.add(Dense(64, activation = ’relu’))
13: model.add(Dense(1))
14: model.compile(loss =0 mae0 , optimizer = ’adam’)
Train the model
15: for each i in epocs do
16: model.f it(Z1, X1, validation data = (v, V), epocs = i)
17: forecast ← model.predict(z)
18: forecast ← object1.inverse transform(forecast)
19: Test it on our univariate sequence.
20: return rmse, mae, mape, rrse
21 : end

In that step, we will define the model and set the parameters needed for our model. We use "ReLu" as activation function. We include Conv1D as a convolutional layer since we are working with a one-dimensional sequence. We then add the Max-Pooling layer and the Flatten layer for clustering and flattening the input matrix that will be used as input to the fully connected neural networks to learn our pattern.

Table 8 1D-CNN Evaluating forecast accuracy for Station

	MAE	RMSE	MAPE	TR%
Product 01	832.40	1138.47	0.067	93.22
Product 02	3103.91	4039.27	0.096	90.4

Analysis of results

The combined results of aggregated mean absolute percent- age error of all the three stations 1, 2, 3 over all days of the week for each of the algorithms are shown in table 9.

Models	ARIMA	Fb Prophet	LSTM	Transformers	1D-CNN
Product01	91.89	90.918	93.96	92.96	93.22
Product02	90.40	88.279	93.03	91.3	90.4
TR% Avg	91.14	89.6	93.50	92.13	91.81

Table 9 Avg TR%

Model selection

We can see that the sequence to sequence models (LSTM, 1D-CNN and Transformers) have the highest accuracy (lowest MAPE across all the products, with an average of 7.52%). This is followed by ARIMA and Fb Prophet (statistical model).

As a result of this work, the constructed recurrent neural network model (a voting model using the three seq2seq model with weight depends on the MAP average as a probability score for each one), this model was chosen for a robust implementation of a forecasting system that would produce periodic forecasts for gas stations.

Safety stock calculation

Once the law of delivery times has been verified (normal law) and the forecast error has been determined, we can calculate the safety stock that absorbs the uncertainty of delivery times and anticipates the variation in future consumption.

Since the service station considers that it is out of stock at 10% of the total capacity of the tank, i.e. at 5000 liters, we will take this value as the origin for the calculation of the safety stock.

The law that generalizes the calculation of the safety stock for a delivery time and a variable demand is the following:

$$\boldsymbol{SS = S_{Min} + z \sqrt{D_L\ \sigma_D^2 + D^2\, \sigma_{DL}^2}}$$

SS : Safety stock ; z : Service coefficient for a 5% risk;

DL : Average delivery time; σDL : Standard deviation of delivery time ;

D: Average demand; σD: Standard deviation of demand.

To account for the error factor in demand forecasting, following Herrin's (2005) approach based on the uncertainty of demand for highly seasonal products, we incorporated this error into the calculation of the standard deviation of consumption as follows:

$$\sigma_{\boldsymbol{D}} = \sqrt{\frac{\sum_i \ e_i^2}{n-1}} \qquad \boldsymbol{e_i = P_i - D_i}$$

ei : The forecast deviation of day i;

Pi: The demand forecast for day i;

Di: The actual consumption for day i; n: Data size;

Reduction of losses due to overstocking

To solve the problems related to overstocking, we propose to adopt an accurate replenishment policy. Given the instability of replenishment times and the variability of daily consumption, we have chosen a policy with a fixed order quantity and a variable periodicity, i.e. determining an appropriate ordering point.

After calculating the safety stock, and determining the distribution law governing the delivery times and forecasting the future demand, we can now determine the reorder point Pc according to the following formula:

$$P_c = SS + D \ \times D_L$$

4 Improvement of the Fuel distribution Network

In this section, the study of this problem is presented in the following manner. First, we begin by describing the current distribution situation. We also describe the assumptions and characteristics of this problem, which allows us to clearly state the problem and then to write its mathematical modeling of MD-MCVRP-TW with min-sum objectives after clustering the demands to the number of depots and turn it to a single depot problem MCVRP-TW, and then reformulate it as the form of reinforcement learning as well as the solving approach used. Finally, we present the experimental results at the end of this section.

4.1 Fuel network distributions

Fuel is a crucial element in today's consumer society, as most, if not all, of our transportation fleet depends on it. This importance has been translated in practice by the multiplication of the points of selling in order to answer at most the majority of the demand when it more and more increasing, but before collecting the invaluable liquid in our pumps, the sales suggest, there is a very important and crucial stage, that of the distribution. This last one has as mission the satisfaction of the consumers' needs through a series of operations.

This definition may seem extremely simple, but when it comes to introducing criteria such as the lowest cost or the shortest time, it quickly becomes complex, whether it is fuel or any other industrial product.

The Objective

The goal is to propose fuel distribution plans that take into account both the demands and the technical and practical constraints, on the one hand, no station will be out of fuel, and on the other hand, the distribution costs (related to fuel consumption, regular and overtime, truck rental, etc.) will be minimized. More precisely, we are interested in a fuel distribution problem in a petroleum logistics chain consisting in determining, over a given horizon, the distribution plan (inventory management and transportation decisions) in a network composed of depots and service stations. It is therefore necessary to determine on this planning horizon:

- The stations to visit.
- The routes and trucks to be used.
- The quantities to be delivered of each product and their allocation to the trucks.

Characteristics

- The truck fleet is heterogeneous.
- The nature of the demand is deterministic.
- The type of transport considered is road transport.
- The duration of use of the truck is fixed.
- The loading and unloading times are determined.
- The delivery times of the customers are fixed by themselves.

The Constraint

The establishment of such a program is subject to a set of constraints that can be classified into three categories:

I. Constraints imposed by the clients

Customer orders are characterized by properties that may be due to operational constraints or customer choices. These properties that are constraints for the dispatcher are:

a- Products and order quantities

Each order is composed by one or more products with quantities recorded at the time of the reception of which the delivery must respect and must split or modify them only under certain conditions.

1. Dissociable or not

It is a property of the order which indicates that the products of the same order can be delivered separately or not.

2. Modifiable or not

It is a property of the order that indicates whether the quantity of a certain product can be changed or not. So, the dispatcher can modify the quantities of the orders to improve the quality of his program while respecting the conditions imposed by the distribution policy.

b- Anticipated time window

Each order must be delivered within a time window that the dispatcher must respect in order not to cause an out-of-stock situation at the customer's, or a return of products in case of early arrival at the station

c- Type of site

Customer sites are classified by type to allow for delivery authorization management. However, a tanker can only deliver to a site if it is authorized to deliver to that site type.

d- Intra-Site Time (IST)

During delivery to a customer site, the tanks spend a significant amount of time before they begin unloading. I.e. This time is either due to parking operations because of the location of the storage tanks, or other operations time.

e- Stored products space

Each customer site has a product storage area where for each product we have:

- Accessibility conditions of the tank categories

- Discharge rate (HL/mn).
- Motor pump need or not

II. Constraints imposed by the tanks and environment(routes)

a- Condition of tanks: Tanks have different types of physical states:

The dispatcher must take into consideration the condition of the tank to assign rotations that verify the operating conditions of that tank. This condition is a consequence of the continuous operation of the tanks. This operation can lead to the deterioration of the tanks, which imposes additional operating conditions. Also, maintenance that improves the condition of a tank may impose additional operating conditions.

CODE	DESCRIPTION
0	Altered
1	Good

b- Category of tanks

There are 3vdifferent categories of tanks:

CODE	DESCRIPTION
1	Small porter
2	Medium truck
3	Big truck

Exp. Some customers have limited sites and therefore require deliveries using "small trucks". A "large truck" category tanker cannot access the customer's site to carry out the unloading operation, which is the operation of placing the product on the customer's site.

c- Filling

At the beginning of the rotation, when the tank is in the depot, it is "empty", i.e. with a fill level of "0", and then, after loading, it becomes "100", i.e. "full", during the deliveries made in the rotation. However, there are some paths that require a certain maximum fill of the tank for it to access them. This means that a tanker can access a path as long as it has not exceeded the filling required by the type of path.

CODE	DESCRIPTION
0	Empty
100	Filled

d- Tank compartmentalization

There are more than 170 different compartmentalization models, each model can have several compartments of different capacities and which can contain different products in the same rotation.

e- Pump or no pump disposition

The unloading of the product at the customer's site is usually done by gravity, as the customer's storage tanks are buried and the product flows by gravity once the piping is connected. Some customers, however, have exposed storage tanks that are not underground and unloading is impossible without a motor pump. For this reason, there are tanks that have motor pumps that allow them to unload at these customers.

f- Cost of transport

The transport of products is carried out both by a fleet belonging to NAFTAL, as well as other fleets of external carriers under contract with NAFTAL for participation in the transport operation.

g- Type of route

It is a characteristic of the road that imposes accessibility conditions on tanks. There are 5 different types: These types of roads require accessibility conditions on the category, condition and filling of the tank.

CODE	DESCRIPTION
0	Easy, accessible to all categories of tanks
1	Difficult, accessible to all categories of tanks
2	Easy, accessible only to small porters
3	Difficult, accessible only to small porters
4	Not accessible to tanks with altered condition

h- Starting point

The dispatcher must take into consideration the starting point of each tanker, because the distribution is dynamic and the tanks are almost continually moving and can start the program execution from any point on the map.

III. Constraints imposed by the depots

a- Hours of service

The depots have different service schedules that must be respected by the dispatcher.

b- Average waiting time

In each depot, tanks are waiting in a line before being loaded. The average waiting time of a tank in a line-up differs from depot to depot. It is included in the calculation of the service time of the tanker rotations.

c- Availability of products in global quantity and by product

Each depot may provide a limited quantity of each available product. The sum of the quantities of each product scheduled to be removed from that depot shall not exceed that availability.

d- Loading speed of each product

In a depot, products can have different loading flows. The loading time of a tanker is calculated according to the loading rates of the products and the capacity of the tanker.

e- Cost of pick up

The costs of picking up products differ from one depot to another. For example, a liter of a product "p" picked up from one depot "x" may cost less than a liter of the same product picked up from another depot "y", taking into account the storage charges of the different depots.

IV. Constraints imposed by the company's distribution policy.

NAFTAL imposes constraints on these dispatchers in the form of operating rules that they must respect in order to preserve the rights of customers, contracted transporters, drivers and company equipment. They also allow for the standardization of the use of transport and storage facilities to ensure better operations.

a- Maximum angle (Alpha) of rotation

For a particular tank, each rotation must not be at an angle that exceeds a certain limiting degree angle (Alpha).

b- Percentage of gaps

At the end of the rotation schedule, there may be rotations that do not use all the compartments of a tanker. For example, a 250 HL tank that is scheduled to deliver an order of 200 HL. This rotation is called “gap rotation".

c- Minimum and maximum order satisfaction percentage (PSPMIN, PSPMAX)

As mentioned above, the dispatcher can make changes to the order quantities (if the order is modifiable), to improve the quality of the program. However, these modifications have minimum and maximum limits that must not be exceeded. For example, if we have PSPMIN = 71% and PSPMAX = 110% then an order of 270 HL must not be more than 297 HL and less than 191.7 HL

Hypothesis

- The stock of each product in the depots and compartments is always higher than the global demand of the customers within the service station cannot be supplied directly from the depots without trucks.
- The tanker trucks are compartmentalized.
- Company trucks are given priority in the assignment.
- The trucks are operational during the whole service time.

4.2 General mathematical model

The goal of this model is to determine, for each planning period, the trips to be taken by each truck and the products to be delivered in order to minimize the sum of the trip costs, based on the work of (Melechovský, 2013) and NAFTAL old model form (BENANTAR) using new case constraint for a Multi depots Multi Compartment Vehicle Routing Problem with Time Window (MD-MCVRPTW).

Problem formulation

Let $G = (D \cup S, A)$ be a directed graph, where $D = (1, \ldots, d)$ is the set of depots, $S = (1, \ldots, n)$ is the set of stations, and $A = \{(i\ ,j)\ i, j \in U \cup V\ \}$ is the set of arcs. Leave station i within a time window $[\alpha i\ , \beta i\ ]$ and duration angle α, also there are H regular working hours and up to H' overtime hours per working day. Particularly, with n + d nodes (customers and depot) represented as $X = \{x^i\}_0^{i=n} \cup$

$\{e^i\}_0^{i=d}$ and $\{e^i\}_0^{i=d}$ denoting the depots nodes, the customer set is assumed to be $' = \{x^i\}_0^{i=n}$.

Each depots node $e^i \in R^3$, i.e., 3 is defined as $\{(s^i , Q^{ip})\}$, where s^i contains the 2-dim location coordinates of node e^i , and Q^{ip} refers to the limit depots quantities and customer node $x^i \in R^4$,4 is defined as $\{(s^i , d^{ip})\}$, where s^i contains the 2-dim location coordinates of node x^i , and d^{ip} refers to its demand (the demand for depot is 0). Here, we take the heterogeneous compartment of trucks with different capacities into account, which respects the real-world situations. Accordingly, let $V = \{v^{iq}\}_0^{i=k}$ represent the heterogeneous fleet of truck compartments, where each element v^{iq} is defined as $\{(Q^{kq})\}$, i.e., the capacity of compartment v^{iq} for the truck k. And let D(x^i, x^j) be the Euclidean distance between x^i and x^j and for simplification, we assume that all vehicles have the same speed f, which could be easily extended to take different values.

Parameters and notations	
S : Set of clients (Stations)	; S = {1,2, . . . , n}.
0 : Depot index	; D = S ∪ {0}.
P : Product set	; P = {1,2, . . . , p}.

K_1	: Set of company trucks.
K_2	: Set of non-company trucks.
k	: Truck's index.
T_k	: Set of truck compartments k.
q $\in T_k$	: Index of compartment.
CCk	: Capacity of the truck k.
Q^{kq}	: Capacity of compartment q for the truck k.
d^{ip}	: Quantity of product p requested by the customer i.
D(x^i, x^j)	: Distance of the between (x^i, x^j) (i, j $\in$ D).
c_{ij}	: Travel cost of the sub-trip (i, j).
c'_{ij}	: Additional cost associated with the use of privet trucks.
f1 (u / km)	: Constant used to calculatec_{ij}
f2 (u / km)	: Constant used to calculatec'_{ij}
α_t	: Earliest departure time for trip t
β_t	: Latest departure time for trip t
α	: Angle of rotations
t_{ij}	: Travel time of the trip (from i to j).
λ_i	: Service time to customer i
s^{ik}	: The start time of service of the customer i by the truck k ($s^{ik} \geq 0$).
Psp_{Max}	: Max Percentage satisfaction
Psp_{Min}	: Min Percentage satisfaction

Decision variable	
x_k^{ij}	1 : if j is visited directly after i in the tour of truck k, otherwise 0
y_k^{ip}	1 : if customer i receives product p by truck k, otherwise 0
z_k^{ij}	1 : if product p is assigned to compartment q of truck k, otherwise 0
a_k^{i}	1 : if truck k is able to visit customer i, otherwise 0

Define the definition domains of the decision variables:

$$x_m^{ij},\ y_k^{ip},\ z_k^{ij},\ a^{ik}\ \in \{0;\ 1\};$$

Objective function

The primary objective of the company NAFTAL, as any beneficial company, is to have a positive return and this by ensuring good distribution planning of its products. We can, therefore, set the objective as being the minimization of the daily cost of delivery of fuels supported by the company and which translates into:

- The best affectation of NAFTAL trucks.
- The reduction of the share of third party carriers (private and public).

So the objective function [14] is to optimize the total cost of the dispatching for each day:

$$Min\left(\sum_{\mathrm{k \in K1}}\sum_{\mathrm{i \in D}}\sum_{\mathrm{j \in D}} c_{ij}\ x_k^{ij} + \sum_{\mathrm{k \in K2}}\sum_{\mathrm{i \in D}}\sum_{\mathrm{j \in D}} (c_{ij} + c'_{ij})\ x_k^{ij}\right)$$

<u>Subject to:</u>

Ensure that, at most, one truck is assigned to the n-th stations on can satisfied it request:

$$y_k^{ip} \le a_k^i\ ,\quad \mathrm{i\ \in\ S, k\ \in\ K, p\ \in\ P}$$

The sum of the entries for all customers served must be equal to the sum of the exits:

$$\sum_{k \in \mathrm{K}} x_k^{ij} = \sum_{k \in \mathrm{K}} x_k^{ji}\ \ \mathrm{, i, j\ \in\ S,}$$

For a given truck m, n-th station not is served until the $n-1$ is served:

$$x_k^{ij} + x_k^{ij} \le 1\ \mathrm{, i, j\ \in\ S, i\ \neq\ j, k\ \in\ K}$$

[14] Obj. fonction Source : Optimisation pour des problèmes industriels de tournées de véhicules, BENANTAR A. 78p

Impose the depot as the starting (j) and ending point(i) of all of all truck trips:

$$\sum_{\mathrm{j}\in \mathrm{S}} x_k^{0j} \leq 1 \text{ and } \sum_{\mathrm{i}\in \mathrm{S}} x_k^{i0} \leq 1 \ \ ; \ \mathrm{k} \ \in \ \mathrm{K}$$

The capacity of any compartment q assigned to product p must be respected:

$$\sum_{\mathrm{i}\in \mathrm{S}} d^{ip} y_k^{ip} \leq Q^{kq}\,, \qquad \mathrm{k} \ \in \ \mathrm{K}, \mathrm{q} \ \in \ \mathrm{Q}_k, \mathrm{p} \ \in \ \mathrm{P}$$

Means that any request for product p must be delivered by a single or multi trucks:

$$\sum_{\mathrm{k}\in \mathrm{K}} y_k^{ip} \leq |E|, \qquad \begin{cases} |E| = 1 \\ |E| = e, \qquad e \neq 1 \end{cases}$$

Note:

$|E| = 1$ the constraint means that it is not possible to split the delivery of the product p, otherwise is possible to split it to **e** *time.*

Each compartment for trips must leave the depot completely filled:

$$\sum_{q\in Q} Q^{qk} x_k^{ij} = \sum_{q\in Q} Q_q x_k^{0j}$$

Each demand for trips must not be more than the limit depots quantities:

$$\sum_{q\in Q_q} Q^{qk} x_m^{ij} \leq Q_0$$

State that the trip departure occurs after the arrival time of the previous trip:

$$d_m^{ij} \leq \sum_{i,j\in X} \left(d_{m-1}^{ij} + \lambda_j x_k^{ij}\right)$$

Ensures the consistency of the tours in terms of time: the departure time of customer i must be less than that of customer j when x_k^{ij}:

$$s^{ik} + \lambda_{\,\mathrm{i}} + t_{ij} \leq s^{jk}$$

If trip t is in position v in the sequence for this truck, its departure times lie within the time window $[\alpha_t, \beta_t]$:

$$\alpha_t x_m^{ij} \leq d_m^{ij} \leq \beta_t x_m^{ij}$$

Maximum Angle of rotations:

$$D\left(x_k^{0i} + \sum_{i,j \in X} x_k^{ij} + x_k^{0j} \right) \leq \alpha$$

Ensures the link between the visit variables y_k^{ip} and the routing variables x_k^{ij}:

$$y_k^{ip} \leq \sum_{j \in D} x_k^{ij} \ , \qquad \mathrm{i} \ \in \ \mathrm{S}, \mathrm{k} \ \in \ \mathrm{K}, \mathrm{p} \ \in \ \mathrm{P}$$

Note: if customer i is not visited by truck k, the constraint sets the variable y_k^{ip} to 0 for every product p.

Mathematical model: The model for a MCVRPTW is described as follows:

$$\begin{cases}
\min \left(\sum_{\mathrm{k} \in \mathrm{K1}} \sum_{\mathrm{i} \in \mathrm{D}} \sum_{\mathrm{j} \in \mathrm{D}} c_{ij} \, x_k^{ij} + \sum_{\substack{\mathrm{k} \in \mathrm{K2} \\ S}} \sum_{\mathrm{i} \in \mathrm{D}} \sum_{\mathrm{j} \in \mathrm{D}} (c_{ij} + c'_{ij}) x_k^{ij} \right) \\
y_k^{ip} \leq a_k^i \ , \quad i \ \in \ S, k \ \in \ K, p \ \in \ P \\
\sum_{k \in K} x_k^{ij} = \sum_{k \in K} x_k^{ji} \ , i, j \ \in \ S, \\
x_k^{ij} + x_k^{ij} \leq 1 \ , i, j \ \in \ S, i \ \neq \ j, k \ \in \ K \\
\sum_{j \in S} x_k^{0j} \leq 1 \ \ k \ \in \ K \\
\sum_{i \in S} x_k^{i0} \leq 1 \ \ ; \ k \ \in \ K \\
\sum_{i \in S} d^{ip} y_k^{p} \leq Q^{kq} \ , k \ \in \ K, q \ \in \ Q_k, p \ \in \ P \\
\sum_{k \in K} y_k^{ip} \leq |E|, \qquad \begin{cases} |E| = 1 \\ |E| = e, e \neq 1 \end{cases} \\
\sum_{q \in Q} Q^{qk} x_k^{ij} = \sum_{q \in Q} Q_q x_k^{0j} \\
\sum_{q \in Q_q} Q^{qk} x_m^{ij} \leq Q_0 \\
d_m^{ij} \leq \sum_{i,j \in X} \left(d_{m-1}^{ij} + \lambda_j x_k^{ij} \right) \\
s^{ik} + \lambda_i + t_{ij} \leq s^{jk} \\
\alpha_t x_m^{ij} \leq d_m^{ij} \leq \beta_t x_m^{ij} \\
D\left(x_k^{0i} + \sum_{i,j \in X} {}_k^{ij} + x_k^{0j} \right) \leq \alpha \\
y_k^{ip} \leq \sum_{j \in D} x_k^{ij} \ , i \ \in \ S, k \ \in \ K, p \ \in \ P
\end{cases}$$

5 Resolution of the problem for Large and Small scale:

Since the problem is NP-hard, a Two-Level solution approach is proposed. It is based up on breaking the problem down into two routing problems. The first one aims to turn it to single depot problem by determining the clusters for each depot (visiting the clusters for vehicle replenishment).

An exact and metaheuristic algorithm is considered to address this sub-problem. Furthermore, the second one is aimed at finding a visiting order of the customers belonging to each cluster. Exact and approximate methods are proposed to tackle this sub-problem.

Figure 13 Problem resolution steps

To solve the problem of np-hard computation in large scale problem we propose the following approach: The steps composing our exacte and heuristic are divided into two phases: A first one is minimizing the total computational number of variable and constraint by clustering, then a second one proposing the hybrid exacte heuristic and DL model using the reinforcement learning of the tours.

Phase I: Preliminary approach

In order to start planning tours, the following operations must be established firstly:

1- **Identification of all the service stations to be supplied during the planning horizon f(t,d,p) for each distribution sector**

Make a list of all the stations having launched an order to be supplied on day **t** and demand **d** for each product **p**,

2- **Determining the planning horizon and the replenishment period**

Determining the planning replenishment period of each distribution cluster in each department “sector of each Wilaya”:

- Make a list of all stations attached to the same cluster distribution center, by specifying the total demand for each product and the geographical position of them (by minimizing the distance between each Depot-cluster and the stations).
- The feasible solutions are composed of routes starting and ending at the depot, which visit each customer exactly once. In this regard, those customers belonging to the same cluster (depot) must be served one after another along the same routes with exact the same vehicles and compartments.

Furthermore, all the routes have to satisfy the truck compartment capacity. This means that the sum of the service demands of all the customers included in a given cluster must not exceed the compartment capacity.

The phase objective is to reduce the computational timing to get a feasible solution by turning the problematics from multi-depots to single-depot.

Using **the clustering problem** the problem in which a given set of weighted objects is to be partitioned into clusters so that the total weight of objects in each cluster is less than a given value (cluster 'capacity') in our case consists of forming a specified number of clusters or groups from a set of elements in such a way that the sum of the demand for each products of the clients in each cluster is within some or less then the product depots capacity limits, and the sum of the distance between the service station and clusters (depots) is minimized.

Given a graph $G = (V, E, P)$ where $\boldsymbol{V}$ is a set of n nodes and $\boldsymbol{E}$ is a set of edges, let $r_{ip} \geq 0$ be the demand request of node client(stations) $i \in V$ for the product $\boldsymbol{P}$ and let $D\ i, j$ be the distance of edge $(i, j) \in E$.

The Clustering Problem consists of partition $\boldsymbol{V}$ into $|k|$ clusters in such a way that the sum of the demand of the elements in each cluster for each products is within some integer capacity limits, "L and U", and the sum of the distance between the elements in the same cluster and it center (depot coordination) is minimized.

The CP can be formulated as a quadratic integer program with binary variables x_{ip}^{j} that take the value of 1 if element (station) i for product p is in cluster j and 0 otherwise.

$$Min \sum_{j=1}^{k} \left(D\ i, j\ \ x_{ip}^{j} \right)$$

Subject to

$$\sum_{j=1}^{k} x_{ip}^{j} = 1, \qquad \forall i \in V, \forall p \in P$$

$$L \leq \sum_{i=1}^{n} \sum_{p \in P} r_{ip}\, x_{ip}^{j} \leq U, \qquad \forall k \in K$$

$$x_{ip}^{j} \in \{0{,}1\}, \quad \forall i \in V, \forall p \in P, \forall k \in K$$

- The first set of constraints forces the assignment of each element to a cluster.
- The second set of constraints forces the sum of all the demand products of elements in the same cluster to be between L and U *(depot limit capacity).*

Example of a feasible solution for a problem instance of the CP dedicated to serving n = 29 clients (stations) organized in $|k| = 3$ clusters.

The objective function adds the total distance of all elements belong to the same cluster.

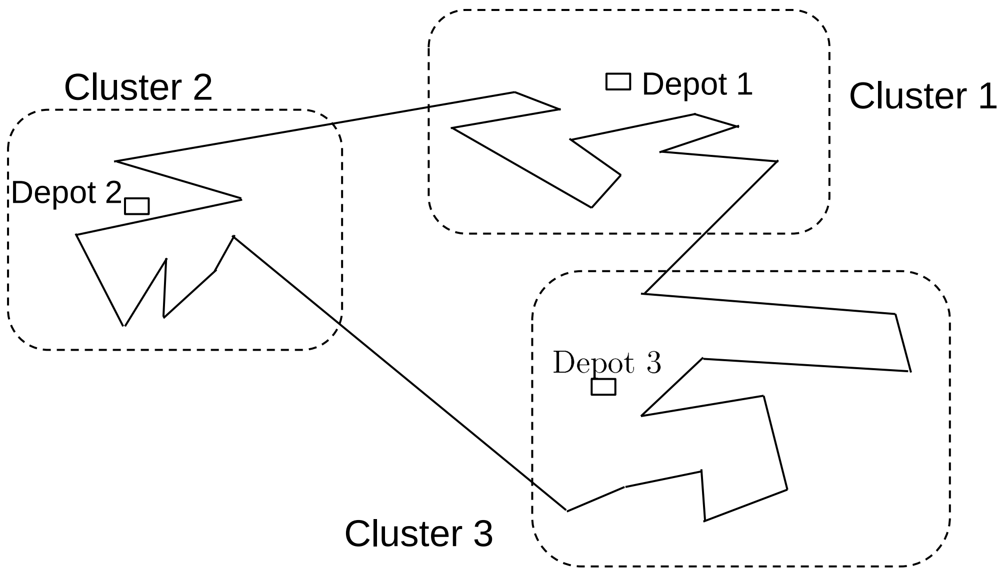


Figure 14 Clustering example

Phase II: Identification of service stations belonging to the sub-clusters

Where a client's demand can be split or not; and supplied by multiple vehicles. The task is to find a set of lowest-cost delivery routes for a fleet of vehicles departing and arriving at the depot, such that each customer belongs to at least one route, each customer's demand is fully satisfied, and the total demand assigned to any (truck) route do not exceed the truck's capacity.

Divides the orders of service stations into sub-cluster (groups) for each distribution center. Each of them constitutes a period **t** and demand **d** for all products. Thus, we obtain a total of sub-periods, which in turn constitute a planning horizon of one day for each sub-cluster .Using two different strategies (homogeneous or heterogeneous clustering).

5.1 The proposed approach

We present our three different approaches to get an optimal or an approximate solution.

I. The exact solution approach

The main objective of this part is to develop an exacte approach to address the MCVRP with time window. To this aim, a mathematical formulation is first developed. However, due to the excessive computational load required to do so, an exacte algorithm is designed to obtain high quality solutions.

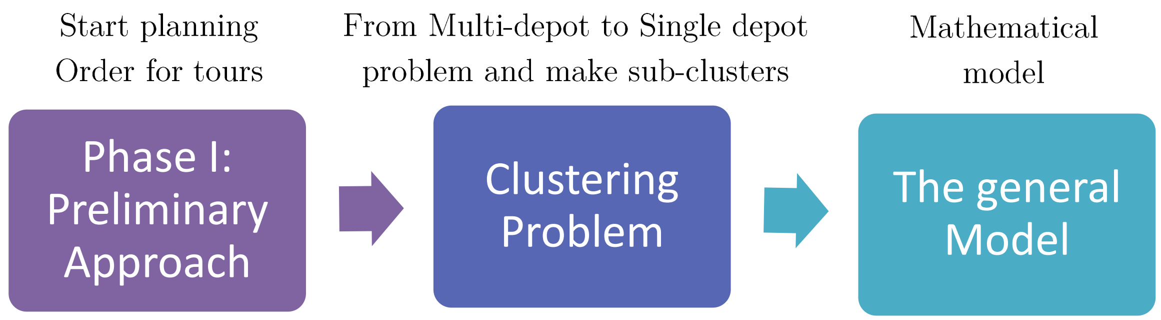


Figure 15 Exact solution approach steps

This optimization technique is actually based on the division of the MCVRP-TW into two different routing problems (by turning the problem form Multi-depot to Single depot problem). The first one aims at identifying the routes to visit the depots clusters. With this objective in focus, an efficient mathematical model is considering. The second problem determines the order of visiting the clients in the same group by divide it into sub-clusters (to reduce the computational timing) and using theme in the general model.

II. The heuristic solution approach

The principle of this heuristic is that each route begins by searching for the nearest unrouted customer to the depot (after the depot clustering phase finished). The proximity relation tries to take into account the geographical and temporal proximity of the customers. For each subsequent iteration, the nearest customer to the last customer added to the route is considered for insertion at the end of the currently generated route. Before adding a customer to the route, the heuristic checks the vehicle capacity constraints and the time window requirement. If the search fails, a new route is started. Note that the insertion values are evaluated in terms of distance and service time. The tankers are assigned sequentially to the list of customers obtained by applying the nearest neighbor constraint procedure, where each customer can be visited once and only once by a tanker k, while checking each time that the tanker capacity constraint is not exceeded; the algorithm of the method is the following:

Step 1. Choose a tanker from the list of available tankers.

Step 2. Start the tour from the depot.

Step 3. Repeat

Find the nearest customer to the last customer on the tour and add it to the tour, by checking the tanker capacity constraint and other constraints specific to the problem;

Until the list of orders is empty.

Step 4. If the list of tankers is not empty, go to (**Step 1**).

Step 5. End Algorithm.

At the end of this step, we obtain a set of routes.

The nearest neighbor constraint procedure:

The advantage of having a list of the neighbors for each customer i is the set of customers that are best to visit immediately after i can be used by the neighborhood search procedure when switching from one particular solution to another and reducing the computational time by masking the client that doesn't satisfy the constraint.

Our approach uses the neighbor-client heuristic as described below, to calculate these values once at the beginning and store the best $N(i)$ customers as a list of neighbors of i.

Neighbors-Customer Heuristic:

1. ***Do*** *: Calculate the arrival time for each i from the actual truck position :*

$$\alpha_i \leq s_i = s_{i-1} + \theta_{i-1} + t_{i-1} \leq \beta_i \ (1)$$

1.1. Search un-served customer i witch satisfy (1) : The least early time ai .

i.e. $\alpha_{min} = Min\{\alpha_i\}, \quad i \in N$;

2. Find the distance between i and the non-masked client:

2.1. $D(i,j) = \sqrt{(xi - xj)^2 + (yi - yj)^2}$.

3. the customers are collocated according to their earliest times ai.

4. Calculate truck-score in i position: N(i) i.e. $N^*(i) = Max\ N(i)$

5. end

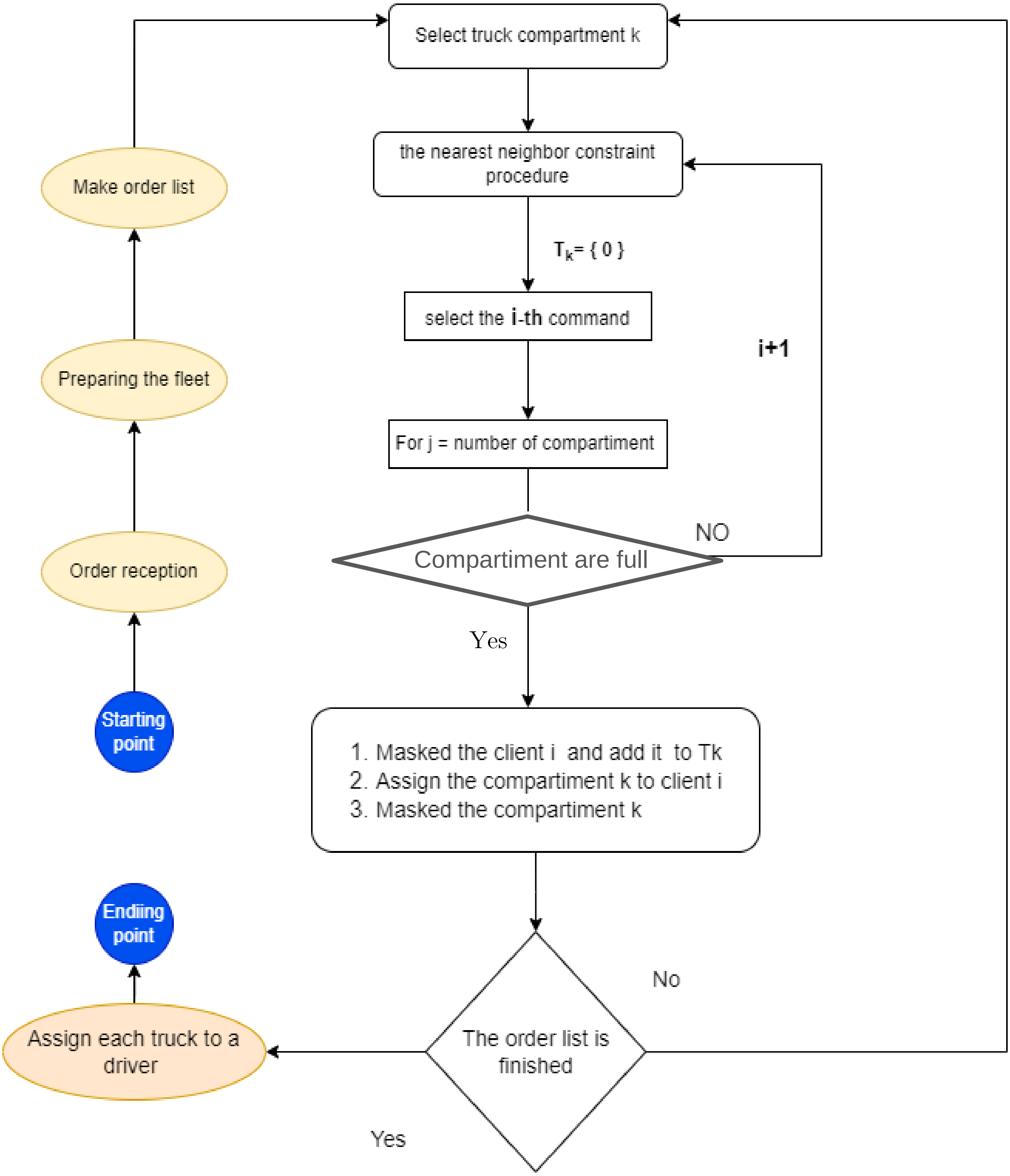


Figure 16 Heuristic Flowchart

III. The Deep Reinforcement learning solution approach

Reinforcement learning (RL) was initially proposed for sequential decision making problems, such as robotics, self-driving cars, etc.

The route construction for MCVRP step by step can also be considered as a sequential decision problem based on the work of Jingwen Li, and Al.[15]

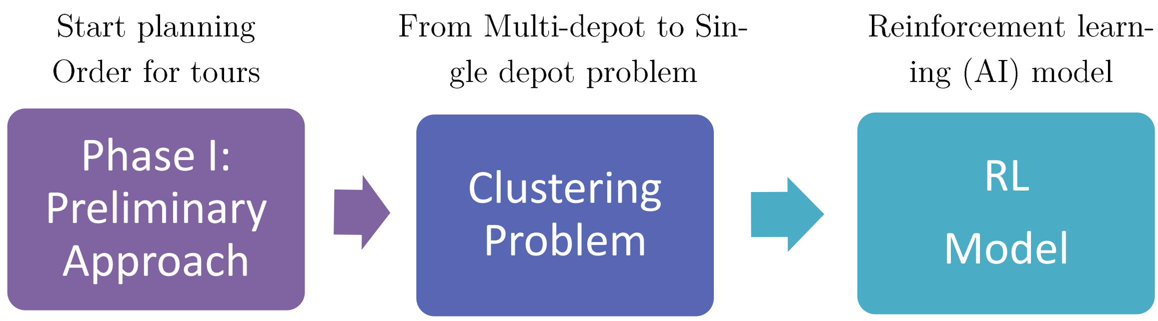


Figure 17 DRL approach steps

Fuel Network Optimization using Deep reinforcement learning

Reformulation as RL Form:

In this part of our work, we define a decision process as a Markov decision process (MDP) defined by 4-tuplets $M = \{S, A, \tau, r\}$ At the same time, the following details of the state space S, the action space A, the state transition rule τ , and the reward function r,are introduced as follow :

State: In our MDP, each state $s_t = (V_t, X_t) \in S$ is composed of two parts:

- The first part of the model is the state of the vehicle Vt, which is expressed as follows:

$$V_t = \{v_t^1, \dots, v_t^m\} = \{(v_t^{11}, \dots, v_t^{1q}),\ (v_t^{21}, \dots, v_t^{2q}), \dots,\ (v_t^{m1}, \dots, v_t^{mq})\}$$

$$V_t = \left\{ \left((o_t^{11}, \dots, o_t^{1q}), D_t^1\ , T_t^1, G_t^1 \right), \left((o_t^{21}, \dots, o_t^{2q}), D_t^2, T_t^1, G_t^1 \right), \dots,\ (o_t^{m1}, \dots, o_t^{mq}\ , D_t^m, T_t^m, G_t^m\ )\right\}$$

where o_t^{im} , D_t^1 and T_t^i represent the capacity of the q compartments remaining, the distance accumulated and the time taken by truck v^i at each step t, respectively. $G_t^i = \{g_0^i, g_1^i, \dots, g_t^i\}$ is the partial route of truck with m compartment at step t, where g_j^i refers to the node visited by the truck v^i at step j .

[15] " Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem., 2021"

Note that the size (dimension) of partial routes (the number of nodes in a route) for all trucks keeps the same and unchanged, i.e., if the truck v^{iq} with the q compartment is selected to serve the node x^j at step t , other vehicles still select their last served nodes. Upon departure from the depot (i.e., t = 0), the initial truck state is set to:" where $Q_{iq}$is the maximum capacity of a compartment v^{im} "

$$V_0 = \{ \ ((Q_{11}, \ \ldots, Q_{1q}),0,0,\{0\}), ((Q_{21}, \ \ldots, Q_{2q}),0,0,\{0\}), \ldots, ((Q_{m1}, \ \ldots, Q_{mq}),0,0,\{0\}) \ \}$$

- The second part is the state of the node S_t, which is expressed as

$$X_t = \{ x_t^0, x_t^1, \ldots, x_t^n \} \ = \ \{(s^0, \lambda_t^0, d_t^{O1}, d_t^{O2}), (s^1, \lambda_t^1, d_t^{11}, d_t^{12}), \ldots, (s^n, \lambda_t^n, d_t^{n1}, d_t^{n2}) \}$$

where s^i is a 2-dim vector that represents the locations , λ_t^i is the replenishment time of the node and d_t^{ip} is a p-dim vector representing the demand of node i for product (in our case p=2) (d_t^{ip} will become 0 once that node has been served for each product). Here, we consider splitting demand, and only nodes with $d_t^{ip} > 0, \forall p \in P$ need to be served.

Action:

We define the action in our method as selecting a truck (by selecting compartment) and a node (a customer or the depot) to visit by filtering the demand characteristics (clients constraints) i.e. the type of demand (existence of auto-pump , split or not split request...) using the filtering condition before any action by the agents (trucks) we matching the agent to the type of the nodes (i.e. r =1 if need the auto pump for the reshipment, 0 otherwise, and masked all the trucks that doesn't match), mask the node if sp =0 otherwise non).

In specific, the action $a_t \in A$ is represented as (v_t^i, x_t^j), i.e., the selected node x_t^j will be served (or visited) by the truck v_t^i at step t.

condition : ($o_t^{ka} - d_t^{jb} \leq maxgap_a \ , if \ a = q \ , b = p$):

"Note that only one compartment and their truck is selected at each step".

Transition:

The transition rule τ will transit the previous state s^t to the next state s^{t+1} based on the previous action a_t :

$$a_t = (v_t^i, x_t^j), \text{ i.e., } s^{t+1} = (V_{t+1}, X_{t+1}) = \tau(V_t, X_t) \text{ and } \quad v_t^i = \{ v_t^{i1}, v_t^{i2}, \ldots, v_t^{iq} \}$$

The elements in truck compartment state v_{t+1}^{iq} are updated as follows:

$$o_{t+1}^{kq} = \{ 0 \quad if \ (o_t^{ka} - d_t^{jp} \leq maxgap_a \ , and \ (k = i \ and \ q = q', b = p) \ \} \quad otherwise \ \ o_{t+1}^{kq} = o_t^k$$

Each demand will retain 0 after being visited:

$$d_{t+1}^{kp} = \{0 \ \ if \ (\ o_t^{ka} - d_t^{jp} \ \leq \ maxgap_a, and \ (\ k = i \ and \ \ q = q' \ , b = p)) \ \}, \quad otherwise \ d_{t+1}^{lp} = d_t^{lp}$$

$$D_{t+1}^{k} = \{ \ D_t^k + D(g_t^k, x^j), if \ k = i \ \} \ , \qquad otherwise \quad D_{t+1}^{k} = D_t^k$$

$$T_{t+1}^{k} = \{ \ T_t^k + \frac{D(g_t^k, x^j)}{speed} + \lambda_t^k, if \ k = i \ \} \ , \qquad otherwise \quad T_{t+1}^{k} = T_t^k$$

where g_t^k is the last element in G_t^k , i.e., last visited customer by compartment of truck v_t^{iq} at step t, and $[\cdot \ , \ \cdot \ , \ \cdot]$ is the concatenation operator. The element in node state $X_{t+1}$is updated as follows:

$$G_{t+1}^{k} = \ \{ \ [G_{t+1}^{k}, x^j] \ \ , if \ k = i \ , \qquad \}, \qquad otherwise \quad G_{t+1}^{k} = [G_t^k, g_t^k]$$

Reward: is defined as the negative value of the total travel distance of all trucks, i.e., $R \ = -\sum_{i=1}^{m} \ \ \sum_{t=1}^{T} \ \ r_t$.

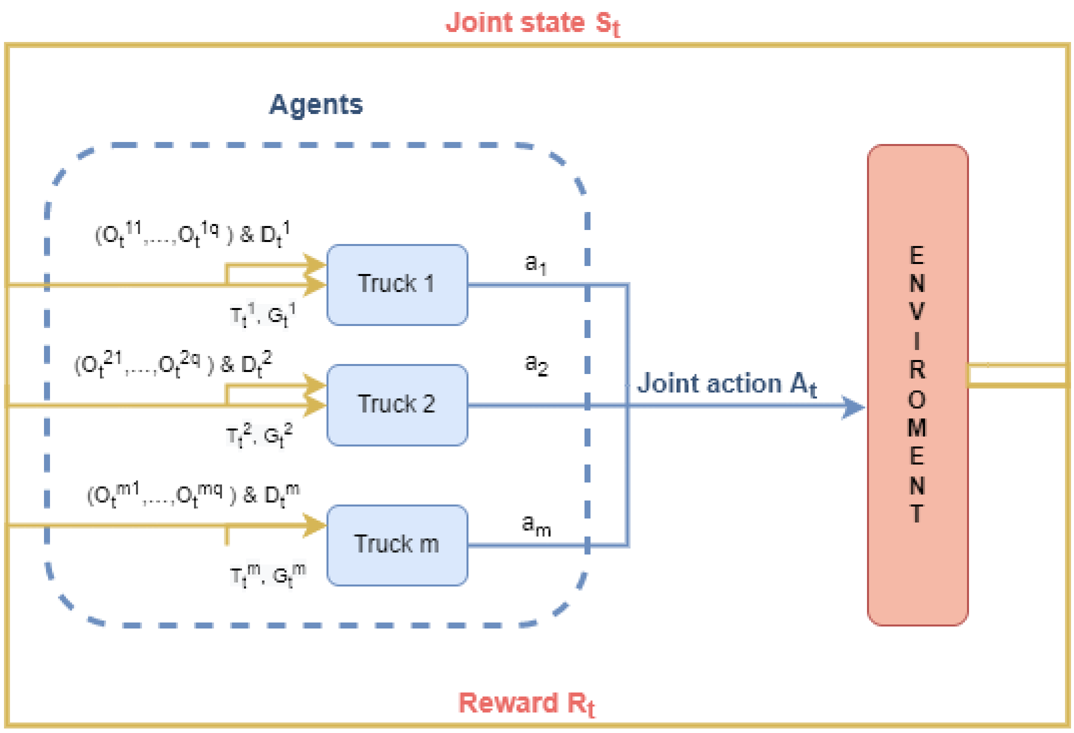
Particularly, assume that node x^j and x^k are selected at step t and $t+1$, respectively, which are both served by the truck compartment v^{iq} ,

Then the reward r_{t+1} is expressed as m-dim vector as follows,

$$\begin{aligned} r_{t+1} = r(s_{t+1}, a_{t+1}) &= r(\ (V_{t+1}, X_{t+1}), (v_{t+1}^{iq}, x_{t+1}^{k} \ ) \ ) \\ &= \{0, \ldots, 0, D(g_t^k, x^j), 0, \ldots, 0\} \end{aligned}$$

where $D(g_t^k, x^j)$ is the distance consumed by the truck v^{iq} for travelling from node x^j to x^k , with other elements in $r(s_{t+1}, a_{t+1})$ equal to 0.

Figure 18 RL flowchart



Framework of Our Policy Network

In our proposed approach, we concentrate on learning a stochastic policy $\pi_\theta(a_t|s_t)$ represented by a deep neural network with a trainable parameter θ. Starting from the initial state s_0 (an empty solution),

We follow the policy π_θ to build the solution by satisfying the MDP until the final state s_f is reached.

$$p(s_f|\ s_0) = \prod_{0}^{f-1} \pi_\theta(a_t|s_t)\ p(s_{t+1}|s_t, a_t\ )$$

Where $p(s_{t+1}|s_t, a_t\ ) = 1$ always holds since we adopt the deterministic state transition rule.

Encoder

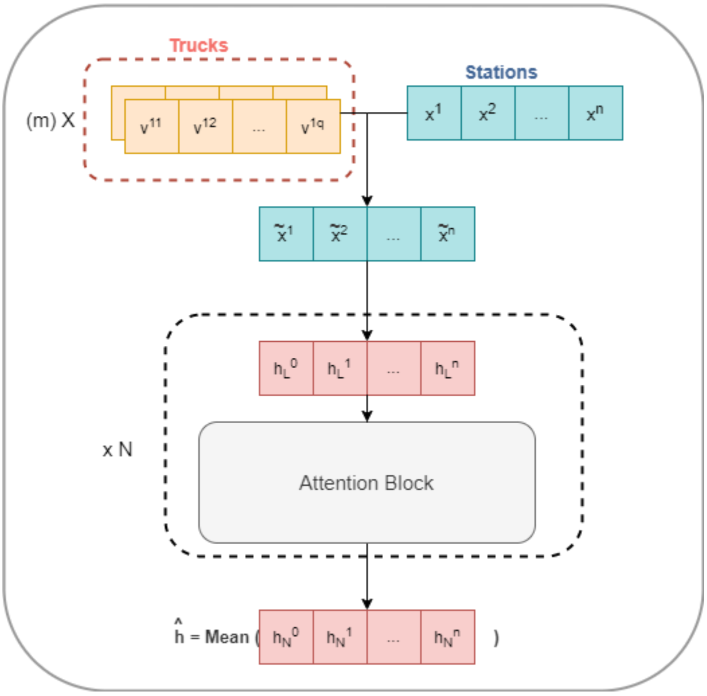


Figure 19 Encoder architecture

The encoder embeds the raw features of a problem instance (i.e., customer location, demand, and vehicle capacity) into a higher-dimensional space, and then processes them through attention layers for better feature extraction. We normalize the demand d_0^{ip} of customer x^i in time step $t = 0$ by dividing the capacity of each vehicle to reflect the differences of vehicles in the heterogeneous fleet, i.e.,

$$\tilde{x}^{\ i} = (s^i, (\ d_0^{i1}/Q_{11}, \ldots, d_0^{i1}/Q_{1q}), \ldots, (\ d_0^{i1}/Q_{m1}, \ldots, d_0^{i1}/Q_{mq})\ )$$

After that, the enhanced node feature $\tilde{x}^{\ i}$ is then linearly projected to h_0^i in a high dimensional space with dimension dim $= 128$.

Afterwards, h_0^i is further transformed to h_N^i through N attention layers for better feature representation, each of which consists of a multi-head attention (MHA) and a feed-forward (FF) sublayer. The l-th MHA sublayer uses a multi-head self-attention network to process the node embeddings $h_l = (h_0^0, h_0^1, \ldots, h_0^n)$. Concretely, we show these steps as follows,

$$Q_{l,y} = h_l\ W_{l,y}^Q, \qquad K_{l,y} = h_l\ W_{l,y}^K\ , \qquad V_{l,y} = h_l\ W_{l,y}^V$$

$$Z_{l,y} = \ Softmax\left(\frac{Q_{l,y}\ K_{l,y}^T}{dim_y}\right) V_{l,y}$$

$$MHA(h_l\ ) = MHA(h_l\ W_{l,y}^Q, h_l\ W_{l,y}^K\ , h_l\ W_{l,y}^V)$$

$$= Concat(Z_{l,1}, Z_{l,2},\ \ldots, Z_{l,Y})\ W_{l,}^O$$

Where $W_{l,y}^Q, W_{l,y}^K \in\ R^{Y \times dim \times dimk}$,$W_{l,y}^V \in\ R^{Y \times dim \times dimv}$, and $W_{l,y}^O \in\ R^{dim \times dim}$ are trainable parameters in layer l and are independent across different attention layers.

Then, the output of the l-th MHA sublayer is forwarded to the l-th FF sublayer with the ReLU activation function to obtain the subsequent hl+1 embeddings. Here, a skip connection and batch normalization (BN) layer are used for the MHA and FF sublayers, which are summarized as follows:

$$r_l^i = BN(\ h_l^i + MHA^i(h_l\ )\ )$$

$$h_{l+1}^i = BN(\ r_l^i + FF(r_l^i)\ )$$

Finally, we define the final output of the encoder, i.e., h_N^i , as the node embeddings of the problem instance, and the mean of the node embeddings, i.e., $h_N = \frac{1}{n}\sum_{i \in X}\ \ h_N^i$, as the graph embedding of the problem instance, which will be reused for multiple times in the decoders.

Decoder

Vehicle selection decoder outputs a probability distribution for selecting a particular compartment of a truck, which mainly leverages two embeddings:

- Truck compartment feature embedding
- Route feature embedding

1) Truck compartment Feature Embedding:

To capture the states of each vehicle at current step, we define the truck feature context $C_v^t \in\ R^{1 \times (q+2)m}$ at step t as follows,

$$C_t^v = [\ (\breve{g}_{t-1}^1, \ldots, \breve{g}_{t-1}^1), D_{t-1}^1, T_{t-1}^1, (\breve{g}_{t-1}^2, \ldots, \breve{g}_{t-1}^2), D_{t-1}^2, T_{t-1}^2, \ldots, (\breve{g}_{t-1}^m, \ldots, \breve{g}_{t-1}^m), D_{t-1}^m, T_{t-1}^m\ ]$$

Where $\breve{g}_{t-1}^1$ denotes the2-dim location of the last node $\breve{g}_{t-1}^i$ in the partial route of truck v^i at step t − 1, and T_{t-1}^i is the accumulated travel time of vehicle v^{ip} till step t−1. Afterwards, the vehicle feature context is linearly projected with trainable parameters $W_1$and b_1 and further processed by a 512-dim FF layer with ReLU activation function to engender the vehicle feature embedding H_v^t at step t as follows,

$$H_v^t = FF(W_1 C_t^v + b_1)$$

2) **Route Feature Embedding:**

The route feature embedding extracts the information from existing partial routes of all trucks, which allows the policy network to learn from the nodes visited in the previous steps, instead of just hiding them. For each truck at step t, we define its route feature context $\check{C}_t^i$ as an arrangement of the node embeddings

(i.e.,h_N^k is the node embeddings for node x^k), corresponding to the node in its partial route G_{t-1}^i . Specifically, the route feature context $\check{C}_t^i$ for each truck compartment v^{ip} , i = 1, 2, ..., m is defined as follows:

Where it was the vehicle i in each step from 0 to t-1

$$\check{C}_t^i = [(\hbar_0^i, \dots \hbar_0^i), (\hbar_1^i, \dots \hbar_1^i), \dots, (\hbar_{t-1}^i, \dots \hbar_{t-1}^i)]$$

Where $C_t^i \in R^{(t\times q)\times dim}$ (the first dimension is of size t×p since G_{t-1}^i should have t elements at step t for q compartment) and $\hbar_j^k$ represents the corresponding node embeddings in h_N of the j-th node in partial route G_{t-1}^i of truck v^{ip} .

Then, the route feature context of all trucks is aggregated by max-pooling and concatenated to give the route context. $\check{C}_t^R$ for the entire fleet, which is then processed by a linear projection using trainable parameters W2 and b2 and a 512-dim FF layer to engender the route feature embedding H_t^R at step t as follows:

$$Cm_t^i = [max(\check{C}_t^i), \dots, max(\check{C}_t^i)] \qquad for \ \ i = 1{,}2, \dots, m \quad Cm_t^i \in R^{1\times q}$$

$$\check{C}_t^R = [\, Cm_t^1, Cm_t^2, \dots, Cm_t^m \,] \qquad \check{C}_t^R \in R^{1\times q\times m}$$

$$H_t^R = FF(\, W_2 \, \check{C}_t^R + b_2 \,)$$

Finally, the truck feature embedding H_t^v and route feature embedding H_t^R are concatenated and linearly projected with parameter W3 and b3, which is then processed by a softmax function to calculate the probability vector as follows:

$$H_t = W_3 \, [\, H_t^v, H_t^R \,] + b_3$$

$$p_t = Softmax(\, H_t \,)$$

Where $p_t \in R^{q\times m}$ and its element p_t^{iq} is the selection probability of the truck compartment v^{iq}at time step t, by getting the maximum probability of the vector p_t. The selected vehicle v^{iq} is then used as input to the node selection decoder.

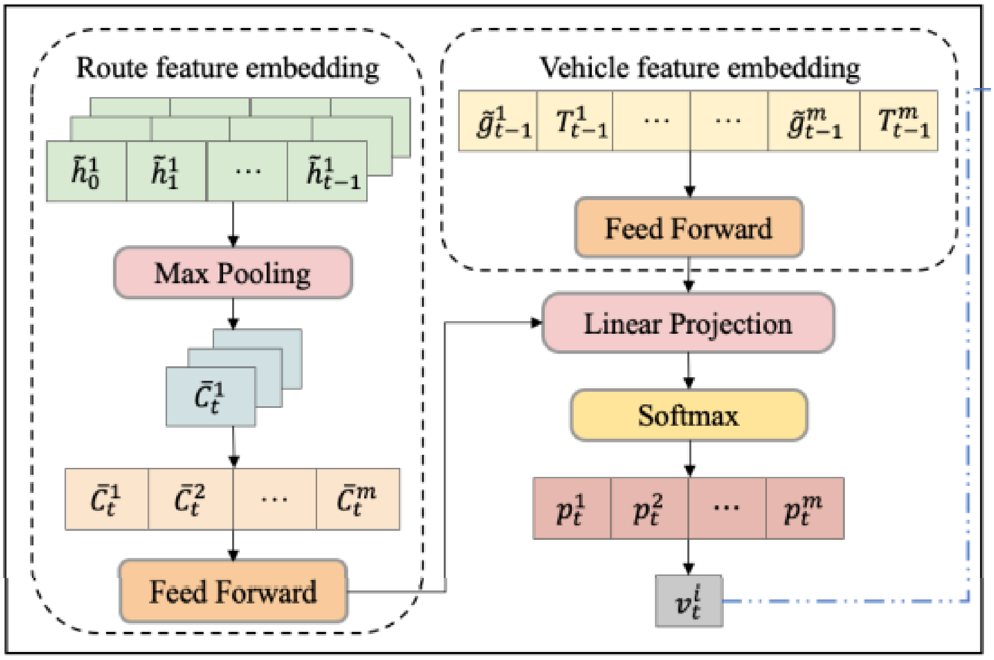


Figure 20 Route and Truck Feature Decoder

Node Selection Decoder:

Considering node embeddings from the encoder and the truck compartment selected v^{iq} from the truck selection decoder, from the truck selection decoder, the node selection decoder produces a probability distribution $\underline{p_t}$ over all unserved nodes (nodes served and satisfied in the previous steps are hidden), which is used to indicate a node that the selected vehicle should visit. For this purpose, we first define a context vector H_t^c as follows:

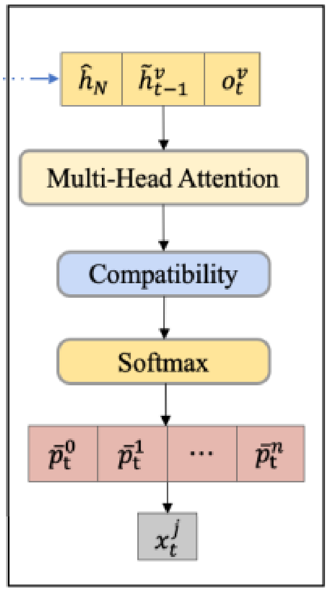
$$H_t^c = [\hat{h}^N, \hbar_{t-1}^i \quad , o_t^{iq}]$$

It consists of the graph embedding $\hat{h}^N$, node embedding of the previous node visited by the selected vehicle, and the remaining compartments capacity of this truck, where the second element $\hbar_{t-1}^i$ is the node embeddings for node, and is replaced with trainable parameters for t = 0.

The context vector H_t^c and the node embeddings $\hat{h}^N$ are then fed into a (MHA) layer to synthesis a new context vector $\hat{H}_t^c$. The query of the attention comes from the context vector, while the key and value come from the node embeddings as shown below:

$$\hat{H}_t^c = MHA(\ H_t^c \ W_c^Q , \ h_N \, W_c^K \ , \ h_N \ W_c^V \ )$$

Where W_c^Q , W_c^K and W_c^V are trainable parameters. We also generate the probability distribution $\underline{p_t}$ by matching the relationship between the enhanced context $\hat{H}_t^c$ and the node embedding h_N through a layer of compatibility. The context compatibility of all the nodes at step t is calculated as follows:



$$u_t = C\ tanh\left(\frac{q_t^T\ K_t}{\sqrt{dim_k}}\right)$$

$$\underline{p_t^j} = Softmax(\ u_t\ )$$

$\underline{p_t^j}$ represents the probability of selecting node x^j served by The selected truck with the compartment v^{iq} at step t .

Figure 21 Node Selection Decoder

Algorithm: MCVRP_RL Pseudocode

```
# R, s, a, t: returns -to -go , states , actions , time
# transformer : transformer with causal masking
# embed_s , embed_a , embed_R : linear embedding layers
# embed_t : learned episode positional embedding
# pred_a : linear action prediction layer
# main model def DecisionTransformer (R , s , a , t ):
# compute embeddings for tokens
  pos_embedding = embed_t ( t )
   s_embedding = embed_s ( s ) + pos_embedding
   a_embedding = embed_a ( a ) + pos_embedding
   R_embedding = embed_R ( R ) + pos_embedding
# interleave tokens as (R_1 , s_1 , a_1 , ... , R_K , s_K )
   input_embeds = stack ( R_embedding , s_embedding , a_embedding )
 # use transformer to get hidden
   states hidden_states = transformer ( input_embeds = input_embeds )
# select hidden states for action prediction tokens a_hidden = unstack ( hidden_states )
 # predict action
    return pred_a ( a_hidden )
# training loop for (R , s , a , t ) in dataloader :
   # dims : ( batch_size , K, dim )
```

```
   a_preds = DecisionTransformer (R , s , a , t )
  loss = mean (( a_preds - a )**2)
   # Getting the (L2) loss for continuous actions optimizer .
   # evaluation loop target_return = 1

DecisionTransformer (R , s , a , t )(predict)
   # append new resulats
   # plot the resualts
```

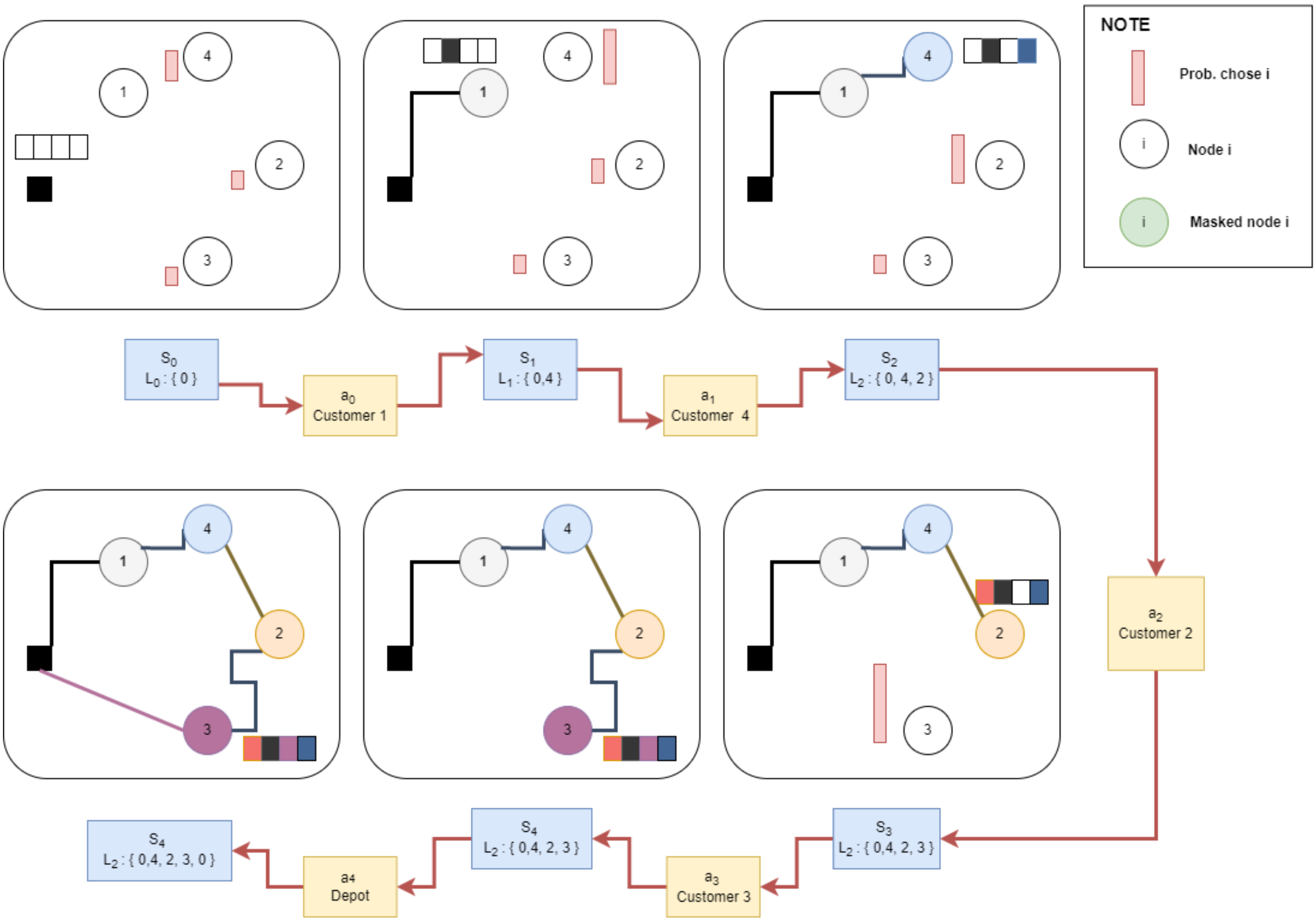


Figure 22 An illustrative MCVRP steps example of a truck with 4 compartments and 4 nodes.

5.2 Experiments and Results

Turning problem from Multi-depots to Single-depot by capacitated clustering

Our CP transforms the original Multi-depots into a Single-depot MCVPR problem into a dimensionally reduced problems (see CP Model). Although the time complexity gain when using the proposed approach is exponential, the solution space sought is reduced, potentially precluding the optimal solution. Station delivery request addresses are converted into 2D-coordinates and are used to compute the distances between delivery points and the depots (the 2D-coordinates center of clusters) .

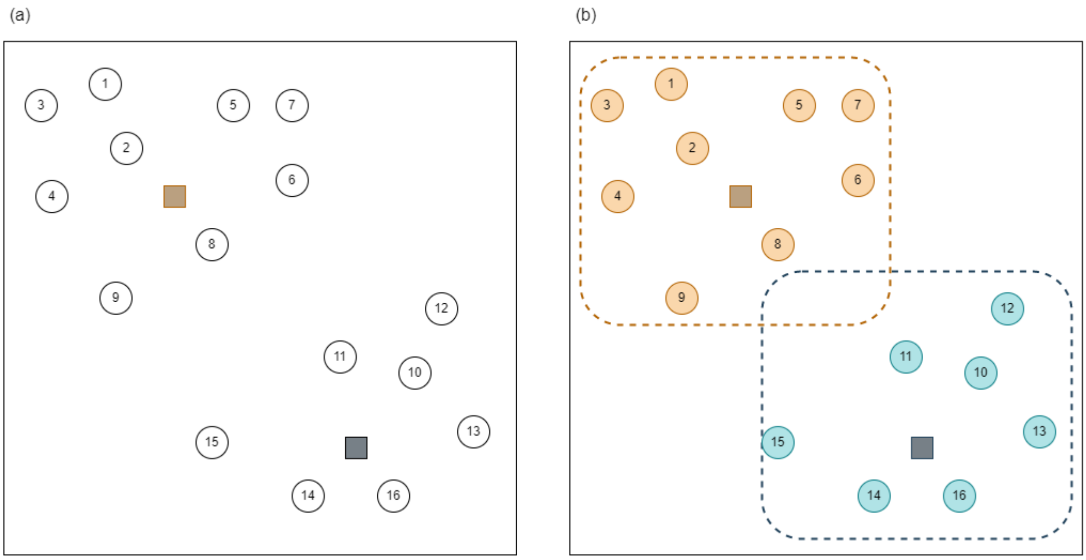


Figure 23 Overview of the CP: MD to SD problem

(A) Initial state with 16 customers and 2 depots.(B) Clustering results in two clusters found.

In the table above we use that data with 4 depots [350 u.q, 400 u.q] to deliver 1458u.q of single product $|P|=1$, as request for 89 stations with [100] x [100] coordinate space.

<table>
<tr><th rowspan="2">Delivery request</th><th rowspan="2">Number of depots</th><th rowspan="2">Loading Rate (%)</th><th rowspan="2">Depot Qi</th><th rowspan="2">Depot Qi max</th><th rowspan="2">$|P|$</th><th colspan="3">Result</th></tr>
<tr><th>Distance*</th><th></th><th>T.CPU</th></tr>
<tr><td rowspan="4">1458</td><td rowspan="4">4</td><td>99.49</td><td>395</td><td>397</td><td rowspan="4">1</td><td rowspan="4">1774.76</td><td rowspan="4">.</td><td rowspan="4">0.39 sec</td></tr>
<tr><td>100</td><td>363</td><td>363</td></tr>
<tr><td>85.9</td><td>322</td><td>375</td></tr>
<tr><td>94.74</td><td>378</td><td>399</td></tr>
</table>

The computational results indicate that our approach have a high performance in clustering of over a wide range of problem instances. In this regard, a sup-cluster generation to define problem instances is presented in the table above:

DEMAND	X-COORD	Y-COORD	READY-TIME	DUE-TIME	SERVICE-TIME
0	35	35	0	230	0
10	41	49	133	198	10

DEMAND	X-COORD	Y-COORD	READY-TIME	DUE-TIME	SERVICE-TIME
7	35	17	22	87	10
13	55	45	98	143	10
19	55	20	123	184	10

The hybrid solution approach example

The two phase heuristic method we propose was programmed in python, and run on a Laptop core i9 with 14 GB of RAM. The performance of the hybird method has been tested on a set of problem cases for the MCVRP problem.

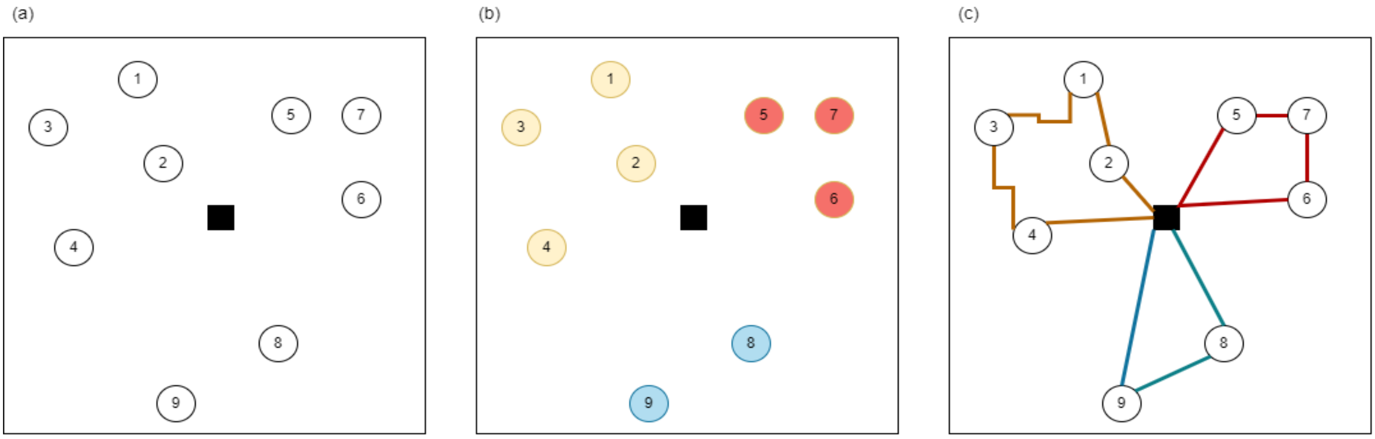


Figure 24 Overview of the MCVRP and the Two-Phase-Heuristic

Step 01: Phase **01** (A) Initial state with 9 customers and 1 depot.
(B) Clustering results in three clusters found.

Step 02: Phase **02** (C) Routing phase determines shortest path inside each cluster using DRL.

Using Single-depot MCVRP instances for the example were randomly generated and used to in the two phases models in the three approaches

Phase I We report it in tables with the following problem parameters:

- Each customer **i** dispose of its demand quantity Q_{ip} for each product p.
- Each cluster is characterized by:
  - Qi The demand requested satisfaction capacity of each depot
  - Qi max The i depot capacity limit
  - |P| Number of products
  - |K| Number of trucks

Table 10 Phase I exemple

Case	|K|	Qi	Qi max	|P|	Result		
					Cost*	Type	T.CPU
C01	2	950	957	1	3316.19	Opt.	0.27 sec
		860	950				
C02		990	993		3314.31		0.36 sec
		820	999				
C03		690	780		1955.5		0.28 sec
		768	773				
C05	4	395	397	1	1774.76		0.39 sec
		363	363				
		322	375				
		378	399				
C06		420	421				
		382	385				
		297	416				
		359	428				
C08	8	196	237	1	1067.88		0.42 sec
		139	230				
		234	238				
		215	238				
		230	239				
		172	239				
		203	239				
		69	230				

In this phase, we group each station to |K| clusters, which can be satisfied by each truck bellowing to same cluster,
i.e. (in case C01 there are 2 trucks with capacity 957 and 950 u.q, to satisfy request of 1810 u.q of 40 clients).

Phase 02 Using the Deep Reinforcement learning model
Using Single-depot MCVRP instances for the example were randomly generated:
(See index 1)

Table 11 Phase 02 DRL example

Depot Cap.	No. Customers	No. Vehicles	Capacities	Cost
150	20	5	30	10.05

Train the model

To train the network, we use familiar policy gradient approaches. To use these methods, we parametrize the stochastic policy π with parameters θ, where θ is the vector of all the trainable variables used in the embedding, decoder and attention mechanism.

$$p(s_f|\ s_0) \ = \prod_{0}^{f-1} \ \pi_\theta(a_t|s_t)\ p(s_{t+1}|s_t, a_t\ )$$

The policy gradient methods use an estimation of the gradient of the expected performance with respect to the policy parameters to improve the policy iteratively.(See index 2)

Table 12 Table of training epochs

Epoch No.	Loss	Mean	Validation score	Cost
01	1.934	11.61	16.71	13.51
02	-75.76	11.61	14.95	13.30
03	-72.741	11.61	13.573	13.23
04	-62.72	11.61	11.44	12.91

Table 13 Depot 01 paths length

Path No.	Nodes	Length
01	depot, 15, 5, 4, 16, 11, depot	2.84
02	depot, 18, 7, 20, 9, 8, 14, depot	2.89
03	depot, 1, 6, 2, 19, 13, 12, 3, depot	2.87
04	depot, 10, depot	0.57
05	depot, 17, depot	0.89
Total	(5 paths)	10.05

Our training methods are quite standard, and due to space limitations:

(i) an actor network that makes predictions about a distribution of probabilities on the next action at any given decision step,

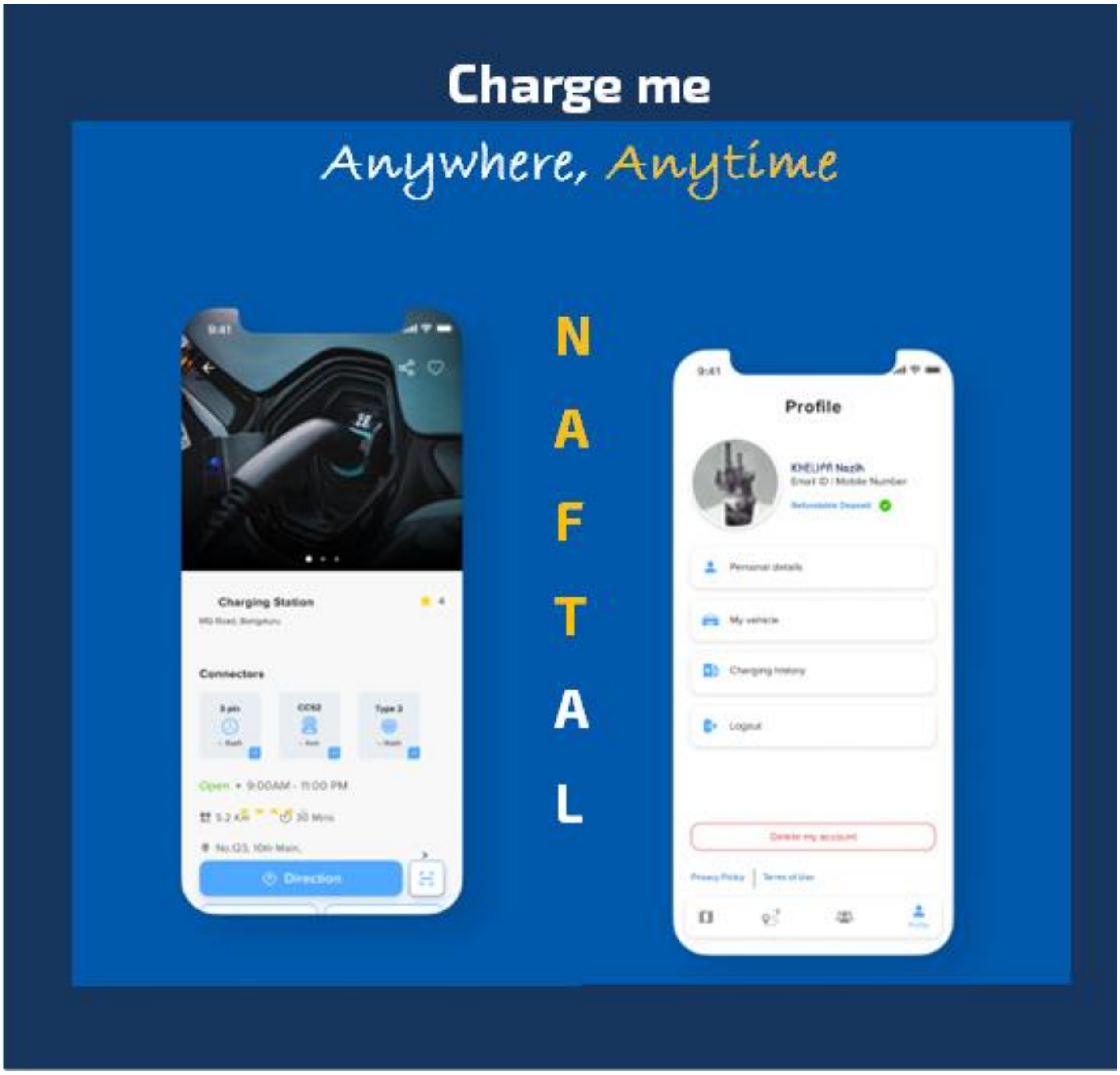
(ii) (ii) a critic network that approximates the reward for any given problem instance from a given state. In our case, we have instance of graph size of 20 nodes with Hypermetres like (5 epochs, 512 samples, Batch of 128 and Embedding_dim of 128 using Learning_rate of 0.00001).

Figure 25 Depot 01 tour paths

The results showed the potential of DRL to generate good policies to solve the problem ‘‘near-optimal solutions’’ of routing a fleet of trucks with heterogeneous capacities and to make this task automatic by finding a global optimal policy that can be applied to arbitrary instances. Furthermore, our proposed method has competitive run times compared to other methods.

In this chapter, we have given a description of the application designed, note that the latter was developed using the programming tools, which allows us to manage the database of customers, tanks, drivers …

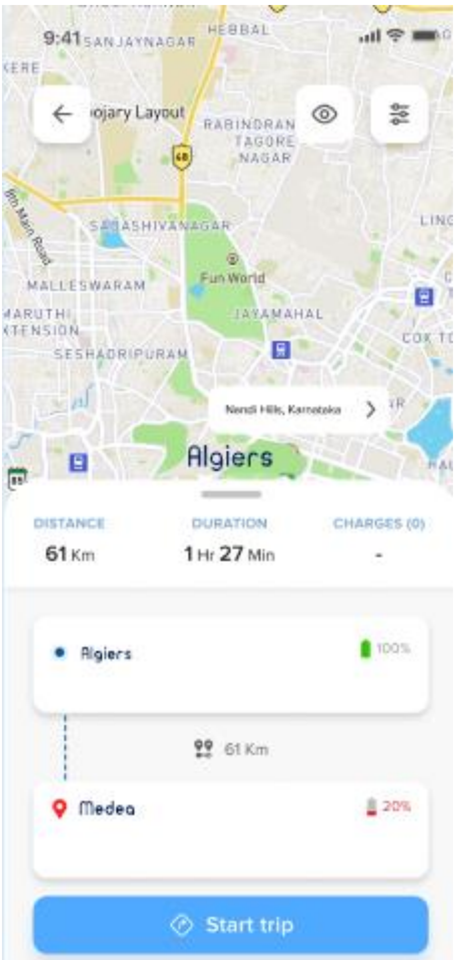
Our solution in action



With 2,276 service stations, the client is the first in the distribution of oil products and related services in Algeria. For that we create the client's application "Charge me" which, by leveraging location-based services, helps to find the closest gas station to the user.
The user can simply click on a station on the map and drive to it.

In another aspect, the mobile application provides information about the client's products, services and available facilities at its service stations.

The NAFTAL upcoming events and current promotions are also available on the app, where they can be reviewed and participated in. Clients can provide feedback to help improve products and services.



The application "Charge me", at the time of registration, stores the user's driver's license. An integrated function of the application is to alert the user when their driver's license is about to be expired.

The challenge increased when we realized that the license expiration date tends to vary from user to user.

Upon registration, the application needed to store the vehicle and the mileage data. Filter the results by region (location) and sub-filter them by (services, features and equipment).

Conclusion

The objective of this chapter is the proposal of solutions and models related to the improvement of the performance of the fuel distribution network and also the internal gas station operation while using IoT technologies. This improvement concerned two main axes of our chain network, namely:

- Demand control: From the sales history, obtained by the data relating to the variation of the stock level in real time of the platform, we first carried out a forecast of the demand, by using statistical and deep learning tools. We obtained a demand with a reality rate of 91.9%, i.e. an error of about 08%.
- Distribution optimization: After optimizing the internal operation of the service stations, we have extended to a global optimization of the fuel supply chain, by focusing on the focal point of the chain, namely the distribution of fuel from the storage centres of NAFTAL. We first proposed a mathematical model by minimizing the transport costs taking into account some constraints related to human, material & time and client resources. Then, an exact method for the resolution of this model by optimizing the planning of rounds of trucks transporting the different types and quantities of fuels, indeed developing another approximate model using clustering and reinforcement learning to deal with the hard computing with large scale problem .

General conclusion

The goal for NAFTAL is the preparation of a daily distribution program for the fleet at the disposal of the center, in order to satisfy the customer orders. This program consists of a set of rounds (rotations) to be built for each tanker of the fleet. This fuel distribution plans take into account both the demands and the technical and practical constraints, on the one hand, no station will be out of fuel, and on the other hand, the distribution costs (related to fuel consumption, regular and overtime, truck rental, etc.) will be minimized. More precisely, we are interested in a fuel distribution problem in a petroleum logistics chain consisting in determining, over a given horizon, the distribution plan (inventory management and transportation decisions) in a network composed of depots and service stations.

Based on the idea that the stations demand forecast is the fundamental point for the planning of all other tasks related to the delivery control. To carry out this diagnosis, the use of IoT technologies was essential for the fuel stations to build a strong and clearly background about the demand historical and obtain the fuel level variation in tanks in real time to make a robust forecasting to reduce the losses related to over and out stocks based on the ability to determine the season and average frequency of replenishment, the daily level of consumption and the average delivery time, we performed a demand forecast with a precision of 92%, and the external management issues linked to the efficiency of its distributor.

In this study also, we have applied distribution network performance techniques at NAFTAL using different mathematical approaches, related to the optimization of fuel distribution, in other words, the planning of tours. We started by translating this problem into a mathematical program (Exact method) and as RL form using the Markov decision process to use it in the Metaheuristic method including an objective function minimizing transportation costs and also the constraints related to the capacities of the trucks and tanks Include the possibility of different capacities.

We propose a learning-based heuristic method that combines deep reinforcement learning and an attention mechanism to learn a route construction policy. In particular, the policy network is based on an encoder, a vehicle selection decoder and a node selection decoder to select a vehicle and a node for that particular vehicle at each step.

Finally, our heuristic models is based on the data for that we generate random locations in a [0,1] square for training and testing, and to finish we need all service stations and distribution centres invest in the installation of this platform(data entry using IoT sensors), the integration of information flows will be total, thus ensuring the synchronization of flows.

.

Bibliography

[1] *A genetic algorithm-based classification approach for multicriteria ABC analysis.* **Kaabi, H., Jabeur, K., & Ladhari, T. 2018.** s.l. : International Journal of Information Technology & Decision Making, 17(06), 1805-1837., 2018.

[2] *Atomic data for resonance absorption lines. III. Wavelengths longward of the Lyman limit for the elements hydrogen to gallium.* **Morton, D. C. 2003.** s.l. : The Astrophysical Journal Supplement Series, 2003, Vol. p. 31, pp. 149(1), 205.

[3] *Erratum to: A branch-and-price algorithm for two multi-compartment vehicle routing problems. .* **Mirzaei, S., & Wøhlk, S. 2017.** s.l. : EURO Journal on Transportation and Logistics, 6(2), , 2017.

[4] *Managing inflexible supply chains.* **Jenkins, G. P., & Wright, D. S. 1998.** s.l. : The International Journal of Logistics Management, 1998, pp. 9(2), 83-90.

[5] *Thirty years of inventory routing. .* **Coelho, L. C., Cordeau, J. F., & Laporte, G. 2015.** s.l. : Transportation Science, 48(1),, 2015, pp. 1-19.

[6] *Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem.* **Jingwen Li, Yining Ma, Ruize Gao. 2021.** s.l. : IEEE Transactions on Cybernetics., 2021.

[7] *Evaluation of time series forecasting models for estimation of pm2. 5 levels in air. .* **Garg, S., & Jindal, H. 2021, ApriL.** s.l. : In 2021 6th International Conference for Convergence in Technology (I2CT), 2021, ApriL, pp. (pp. 1-8).

[8] *Multicommodity vs. single-commodity routing. .* **Archetti, C., Campbell, A. M., & Speranza, M. G. 2014.** s.l. : Transportation Science, 50(2), 2014, pp. 461-472.

[9] *NeuroLKH: Combining deep learning model with lin-kernighan-helsgaun heuristic for solving the traveling salesman problem.* **Xin, L., Song, W., Cao, Z., & Zhang, J. 2021.** 2021, pp. Advances in Neural Information Processing Systems, 34, 7472-7483.

[10] *Supply-chain management in the oil industry. .* **Lisitsa, S., Levina, A., & Lepekhin, A. 2019.** s.l. : In E3S Web of Conferences. EDP Sciences, 2019, pp. (Vol. 110, p. 02061).

[11] **Archetti, Bianchessi, and Speranza . 2015.** 2015.

[12] *Attention Is All You Need.* **Ashish Vaswani, Noam Shazeer.**

[13] **Auto Arima.**

[14] **B. Lindemann , B. Maschler, N. Sahlab.** *A Survey on Anomaly Detection forTechnical Systems using LSTM.*

[15] **Bellman, R. 1957.** *A Markovian Decision Process.* s.l. : Journal of Mathematics and Mechanics., 1957.

[16] **BENANTAR, Abdelaziz.** *Optimisation pour des problèmes industriels de tournées.*

[17] **Bengio, et al. 1994.** 1994.

[18] **Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. 2009.** *. A heuristic for the multi-period petrol station replenishment problem.* s.l. : European Journal of Operational Research, 191(2),, 2009. pp. 295-305.

[19] **Cornillier, F., Laporte, G., Boctor, F. F., & Renaud, J. 2008.** *The petrol station replenishment problem with time windows. .* s.l. : Computers & Operations Research, 36(3), 2008. pp. 919-935.

[20] **Elbek, M., & Wøhlk, S. 2016.** *A variable neighborhood search for the multi-period collection of recyclable materials.* s.l. : European Journal of Operational Research, 249(2), 2016. pp. 540-550.

[21] *Forecasting at scale.* **Taylor SJ, Letham B. . 2017.** 2017.

[22] *Forecasting at Scale: How and Why We Developed Prophet for Forecasting at Facebook.* **Lander Analytics.**

[23] *Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning. .* **Li, J., Xin, L., Cao, Z., Lim, A., Song, W., & Zhang, J. 2021.** s.l. : IEEE Transactions on Intelligent Transportation Systems, 23(3), 2306-2315., 2021.

[24] **Hochreiter & Schmidhuber. 1997.** 1997.

[25] **Hochreiter[German]. 1991.** 1991.

[26] **Lahyani, R., Coelho, L. C., Khemakhem, M., Laporte, G., & Semet, F. 2015.** *A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. .* s.l. : Omega, 51, 2015. pp. 1-10.

[27] **Melechovský, J. 2013.** *A variable neighborhood search for the selective multi-compartment vehicle routing problem with time windows.* s.l. : Lecture notes in management science, 5, 159-166., 2013.

[28] **Neo Wu, B. Green. 2020.** *Deep Transformer Models for Time Series Forecasting.* s.l. : International Conference on Machine Learning, 2020.

[29] **Popović, D., Vidović, M., & Radivojević, G. 2012.** *Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. .* s.l. : Expert Systems with Applications, 39(18), 2012.

[30] **Rabbani, M., Farrokhi-asl, H., & Rafiei, H. 2016.** *A hybrid genetic algorithm for waste collection problem by heterogeneous fleet of vehicles with multiple separated compartments. .* s.l. : Journal of Intelligent & Fuzzy Systems, 30(3), 2016. pp. 1817-1830.

[31] *Real-time dispatch of petroleum tank trucks. Management science,.* **Brown, G. G., & Graves, G. W. 1981.** 1981, pp. 27(1), 19-32.

[32] *Supply-chain management in the oil industry .* **Svetlana Lisitsa,Anastasia Levina and Aleksander Lepekhin. 2019.** s.l. : E3S Web of Conferences, 2019.

[33] *The Prophet on Walmart — Comprehensive Intro to FbProphet.* **Robson, Winston A. 9 July 2019.** 9 July 2019, Medium, Future Vision.

[34] **Trevor Hastie and Robert Tibshirani. 1986.** 1986.

[35] **Vidović, M., Popović, D., & Ratković, B. 2014.** *Mixed integer and heuristics model for the inventory routing problem in fuel delivery.* . s.l. : International Journal of Production Economics, 147, 2014. pp. 593-604.

[36] **Wrobel, A. 1984.** *On Markovian Decision Models with a Finite Skeleton.* s.l. : Mathematical Methods of Operations Research , 1984.

Appendix

Appendix 01

(i) Generate Data for RL approach

Deep learning data analytics uses algorithms that continually improve over time, but quality data is essential for these models to work effectively using torch extensions, numpy, os and pickle library to save the data. For this reason we generate our data from: generate_mcvrp_data() using three main parameters:

- Size of data
- Size of our MCVRP
- Number of vehicles and its compartments.

Fleet data: This is essentially the number and capacity of the trucks, the truck code, the maximum time of use of the trucks, the travel speeds, the fixed and variable costs of the trucks, etc.

Customer data: The most important information is the number of products to be delivered, the quantity of each product, the name and location of the customers, the customer-to-customer distances, the depot-to-customer distances and the estimated loading/unloading times.

Route data: Some route parameters are also needed for dispatching, such as traffic density, constraints on the use of certain routes (bridges, tunnels, etc.) by each truck.

Data on the depot: An estimate of the waiting and loading times at the loading station is also important

```
import os
import numpy as np
from utils.data_utils import check_extension, save_dataset
import torch
import pickle
import argparse

def generate_mcvrp_data(dataset_size, mcvrp_size, veh_num,vhe_comp):
    data = []
    for seed in range(24601, 24611):
        rnd = np.random.RandomState(seed)
         loc = rnd.uniform(0, 1, size=(dataset_size, mcvrp_size + 1, 2))
         depot = loc[:, -1]
         cust = loc[:, :-1]
```

```
        d = rnd.randint(1, 10, [dataset_size, mcvrp_size+1])
        d = d[:, :-1]  # the demand of depot is 0, which do not need to generate here
#example of generating vehicles and it's compartments
         if veh_num == 3 and veh_comp == 2:
             cap = [20., 25., 30. 20., 25., 30.]]
              thedata = list(zip(depot.tolist(),  # Depot location  cust.tolist(),
               d.tolist(),  np.full((dataset_size, 3), cap).tolist()   ))
data.append(thedata)
```

Decoder in code:

```
# Calculate queries, (n_heads, batch_size, n_query, key/val_size)
Q = torch.matmul(qflat, self.W_query).view(shp_q)

# Calculate keys and values (n_heads, batch_size, graph_size, key/val_size)
K = torch.matmul(hflat, self.W_key).view(shp)
V = torch.matmul(hflat, self.W_val).view(shp)

# Calculate compatibility (n_heads, batch_size, n_query, graph_size)
compatibility = self.norm_factor * torch.matmul(Q, K.transpose(2, 3))
```

```
self.n_heads = n_heads
 self.input_dim = input_dim
 self.embed_dim = embed_dim
 self.val_dim = val_dim
 self.key_dim = key_dim
 self.norm_factor = 1 / math.sqrt(key_dim)  # See Attention is all you need

    self.W_query = nn.Parameter(torch.Tensor(n_heads, input_dim, key_dim))
    self.W_key = nn.Parameter(torch.Tensor(n_heads, input_dim, key_dim))
    self.W_val = nn.Parameter(torch.Tensor(n_heads, input_dim, val_dim))
```

<u>Encoder part of code to generate , hi_hat and hi :</u>

```
class MultiHeadAttentionLayer(nn.Sequential):
# multihead attention -> skip connection, normalization -> feed forward -> skip
connection, normalization
    def __init__(
            self,
```

```
            n_heads,
            embed_dim,
            feed_forward_hidden=512,
            normalization='batch',
    ):
        super(MultiHeadAttentionLayer, self).__init__(
            SkipConnection(
                MultiHeadAttention(
                    n_heads,
                    input_dim=embed_dim,
                    embed_dim=embed_dim
                )
            ),
            Normalization(embed_dim, normalization),
            SkipConnection(
                nn.Sequential(
                    nn.Linear(embed_dim, feed_forward_hidden),
                    nn.ReLU(),
                    nn.Linear(feed_forward_hidden, embed_dim)
                ) if feed_forward_hidden > 0 else nn.Linear(embed_dim, em-
bed_dim)
            ),
            Normalization(embed_dim, normalization)
        )
```

Appendix 02

(ii) Training graph

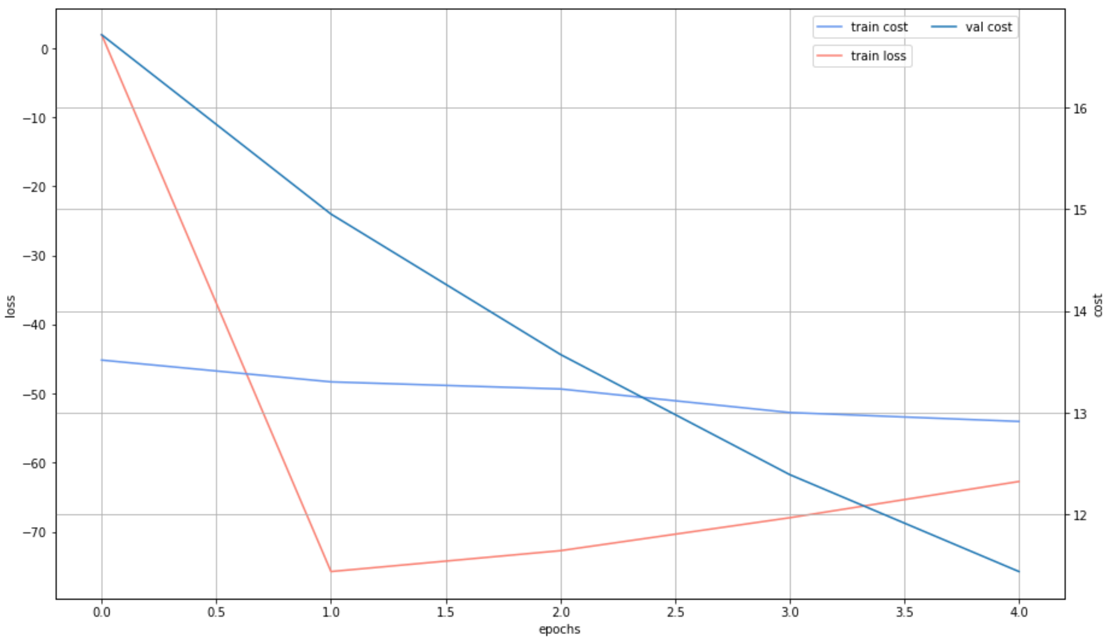


Figure 26 Training graph epochs