Inventory Routing Problem description for ROADEF/EURO 2016 Challenge

Abstract: This document describes the model scope of the Air Liquide Inventory Routing Problem related to the distribution of bulk gases for the ROADEF/EURO 2016 Challenge. It provides a formulation of the optimization model and an explicit formulation of various business and technical constraints that should be respected and how the objective function is defined and computed for a given solution.

Keywords: Inventory Routing Problem, Industrial Gas Distribution

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Context

Worldwide, Air Liquide operates Air Separation Units (ASUs) to produce pure Oxygen, Nitrogen and Argon products through separation from air. Air Liquide serves approximately 1 million customers worldwide in various industries, including healthcare, pharmaceuticals, food & beverage, electronics, welding & cutting, steelmaking, petrochemical & refining, and oil & gas production.

In this ROADEF/EURO Challenge, we will focus on example contexts from our Healthcare business (https://www.airliquide.com/healthcare) which deliveries large (bulk) volumes of liquid oxygen to 7500 hospitals worldwide. On-site vaporization units convert this cold, liquid oxygen into gaseous form (at around room temperature) which travels through the hospital in dedicated piping to supply patients with the oxygen they require.

For bulk delivery customers, Air Liquide installs on-site storage vessels ("tanks"), which are regularly refilled by trucks (*"vehicles"*) transporting the gases as cryogenic liquid ("liquid gas") from our ASUs to our customers.

Through its remote telemetry system, Air Liquide monitors customer tank levels and consumption rates and forecasts each customer's future consumption over the coming hours and days. Through this process, Air Liquide takes the responsibility to guarantee that sufficient inventory of product will be maintained on-site at the customer to meet their demand (in logistics, we call this policy Vendor Managed Inventory, VMI).

While VMI customers are in the majority, a smaller but still significant set of customers (called "call-in" customers) are supplied on an "on demand" policy, as they directly place orders with Air Liquide when they need product.

In this VMI context, Air Liquide must efficiently organize the safe and reliable roundtrips of its truck fleet.

Therefore, on a daily basis, Air Liquide' dispatchers review the forecast of each customers' consumption and then adapt the schedule of the transportation accordingly. Their goal is to reduce the cost per delivered unit (e.g., €/kg) over the long term, while avoiding product shortage (run-outs) , satisfying the orders of the call-in customers and respecting safety and regulatory constraints (e.g., limits on continuous driving time).

Air Liquide's dispatchers have strong experience with this process. However, while it is relatively easy to find feasible schedules avoiding run outs, obtaining the minimum cost per delivered unit is a challenge. The specific complexity of the problem is mainly linked to two factors:

- ➔ There is an exponential number of feasible solutions, making the problem hard to solve even for a few customers;
- ➔ Each short term decision impacts the future cost; indeed, a decision that can reduce the cost today may be not a good decision over the long term.

1.2. Position of the problem in the OR world

In the OR world, this problem can be classified as an "Inventory-routing problem" (IRP), first proposed by Bell et al in 1983. While many mathematical formulations already exist, along with a growing scientific literature, the Air Liquide IRP has several features that make it unique:

- the objective is rational: the goal is to minimize the logistic ratio (e.g., cost per unit delivered: €/kg) which represents the best use of each km driven and each hour spent and not the total number of km or total hours only.
- Secondly, the solution must satisfy specific business-related constraints that that are generally not taken into consideration in the literature.
  - Consideration that drivers' Bases and sources of product are not always co-located: classical models assume that the drivers' Bases and sources are co-located.
  - No prior assignment of drivers to trailers.
  - Accurate modeling of time: Air Liquide IRP model considers an effectively continuous time (accurate to the minute) for the timing of operations (and discrete time for inventory control)
  - Multi-trip problem: a shift may be composed by several trips, i.e. it can alternate loading and deliveries
- Finally, the average size of the problem is bigger than those generally studied in literature.

2. PROBLEM SCOPE

The context described in the challenge is adapted from the Healthcare business environment for the supply of hospitals with bulk liquid oxygen.

The main features of this **Version 1 core model** are:

Product and sourcing: *A Single Product with Unlimited Supply*
- One product: Liquid Oxygen at cryogenic temperature (approximately -220°C)
- One single production site (source) with unlimited product available 24/7 (no capacity constraints on production and storage)
- Safety first: fixed loading times at sources to be able to safely handle cryogenic products

Customers: *Do not leave our customers breathless!*
- Vendor Managed Inventory (VMI) customers only, available 24/7 for delivery (no orders / call-in customers).
- Customer Consumption forecast is known in advance for the whole horizon at an hourly time step.
- Safety levels inside the cryogenic tanks are determined for the whole horizon to always guarantee high enough level of oxygen. No run-out are allowed below safety level.
- Safety first: fixed unloading operation time per customer to be able to safely access site and perform all tasks.

Transportation resources: *Pairing Drivers and Tractor-Trailers*
- All the transportation resources are located at one single Base (which is located at a different location to the production source)
- Several drivers with different availability time windows (Day and night drivers).
- A driver can only drive one trailer (typically each trailer are used by both day and night drivers)
- Heterogeneous Fleet: Each trailer has a specific capacity.

Shift definition: *Happy Drivers, Safely Home Each Day*.
- Each shift has only one driver and only one Trailer.
- A day or night driver has to start from the Base and come back to the Base
- Driver will be paid from the beginning of the shift at the Base to the end of the shift (within the drivers availability hours)
- Safety first: abide by legal regulations, particularly limitations on cumulative driving time

Objective function: *Meet customer demands using time and km at best*
- The objective is to minimize the Logistic Ratio, which is equal to the time and distance cost divided by the total quantity delivered over the whole horizon (€/km).The time cost is proportional to the duration of the shifts (mainly related to the salary of the driver), and includes the driving time, the idle time, and the loading/unloading times. The distance cost is proportional to the distance traveled by a vehicle (mainly related to the fuel consumption).

Instances sizes

- Instances are within a small/medium scale range: 1-4 trailers/drivers, 50 to 200 Customers, Time Horizons from 1 week up to 1 month

Version 2 is an extension of the version 1. This means that we can translate all the instances of version 1 in version 2 with some extension on the master data.
The aim of the version 2 is to approach the problem to the reality of dispatching within Air Liquide, by integrating some very common real-life constraints.

The **additional** features of version 2 are:

Product and sourcing: *A Single Product with Unlimited Supply on multiple sources*
- *Multiple* production sites (and not only one as the V1), which means that any vehicle has the choice to load the product on different sources. Each source has a list of vehicles allowed to load.

Customers:
- *Each customer has opening hours:* The delivery of customer may happen only on certain time windows, corresponding to the opening hours of the customer facility.
- *Call-in customer :* Some customers ask Air Liquide to deliver him with a determined quantity of gas and within a certain time windows. For such customers, there is no need of care about product levels, but the delivery is a must-do.
- *Minimum Operation Quantity:* for each delivery*,* the quantity delivered must be above a certain value, in order to avoid short sighted decisions, by imposing a maximum delivery frequency.
- *Layover customers (call in or VMI):* The layover enables the driver to travel for an extended duration, covering a larger area. A number of customers cannot be served because they are too far from the base. When serving those customers, the driver must include a rest time (layover) into the shift. Each layover has a fixed cost, covering the hotel fare. One layover maximum per shift.

Transportation resources:
A driver can drive multiple trailers, and not only one as the V1

Instances sizes
- Instances are within a medium/big scale range: 1-8 trailers/drivers, 50 to 300 Customers, Time Horizons from 1 week up to 1 month

In the entire document, the differences between V1 and V2 specifications are underlined in blue.

3. PROBLEM SETTING

The problem consists of planning bulk distribution shifts in order to minimize total distribution cost over the long term.

The goal is to build delivery shifts to match the demand requirements subject to given resources and technical constraints in order to **minimize the ratio between the total distribution cost dividing the total quantity delivered over a fixed horizon (*logistic ratio*).**

3.1. Key definitions

Base	A base is both the starting and the ending location of shifts
Vehicle	A vehicle is the combination of a driver and a trailer.
Delivery	A delivery corresponds to a stop of a vehicle at a customer site: the delivery hose is hooked up to the vessel and product is delivered.
Load	A load corresponds to a stop of a vehicle at a source site, where the Vehicle is filled with products.
Shift	A shift is a chronological list of activities made by a driver during his working period. It starts and ends at the driver's BaseFor example, activities would be structured such as "arrive at Base, source, or customer."

Layover	A layover is a fixed idle time interval in a shift which happens when one or more “layover customers” are delivered. The layover enables the driver to travel for an extended duration, covering a larger area. One layover maximum per shift

3.2. Units of measure for quantity

For quantities, we use the weight (Kg).

3.3. Time representation and horizon

Let us consider the scheduling horizon T. We use two discrete time breakdowns of the interval [0, T]: H (Hours) and M (Minutes). Hourly timesteps give the inventory management granularity while minute timesteps give the time granularity for drivers and customers.

3.4. Drivers master data

The set of drivers is referred to as **DRIVERS**. By convention, indices referring to drivers will be denoted by *d*. It represents a driver with his/her characteristics. Each driver *d*∈**DRIVERS** is defined by:

- TIMEWINDOWS(*d*) : the set of availability intervals of driver *d*, each included in [0, T[(in minutes)
- TIMECOST*(d)*: the cost per working time unit of driver *d* (in €/minute). “Working time” means that this cost doesn’t apply during the layover time interval.
- MAXDRIVINGDURATION*(d)*: the maximum driving duration for driver *d*, before ending the shift at the base or doing a layover (in minutes)
- MININTERSHIFTDURATION*(d)*: the minimum time interval for driver *d* between 2 shifts (in minutes)
- TRAILERS*(d):* the set of **trailers** which can be driven by the driver *d*.
- LAYOVERCOST(d): the cost of a layover (in €).
- LAYOVERDURATION(d): the duration of a layover (in minutes), which is fixed. Note that by constraints, any travel that lasts more than Layoverduration+drivingtime must have a layover.

3.5. Trailers master data

The set of trailers is denoted as **TRAILERS**. By convention, indices referring to a trailer will be denoted by *tl.* Each trailer *tl* ∈ **TRAILERS** is defined by:

- DISTANCECOST (*tl*): the cost per distance unit for trailer *tl* (in €/km).
- CAPACITY *(tl*): the capacity of traile*r tl* in mass (Kg). This capacity corresponds to the usable capacity which is the quantity of product that can be loaded in the trailer and delivered to customers. It does not include the minimum quantity of product that must remain in the trailer (to maintain a head of liquid in order to operate the pump) and the volume (i.e., vapor space) that must remain empty for pressure considerations.
- INITIALQUANTITY (*tl*): the mass (in Kg) of usable product in trailer *tl* at time 0. Must belong to the interval [0, CAPACITY (*tl*)].

3.6. Locations master data

A Location denoted *p* (or alternatively *q* when the distance is calculated between two locations) may be a base, a source, or a customer:

- **Bases** are the starting and ending locations of any shifts.
- **Sources** and **Customers** are loading and delivery locations respectively. For sources, quantities loaded to the trailers have a negative sign, while for customers, delivered quantities have a positive sign, as do their consumption forecasts.

Since a part of the customers considered in this problem is VMI customers (forecastable customers), Air Liquide forecasts their tank level and decides to deliver product and whenever necessary and cost efficient.

By convention, the numbering of the locations *p* starts with the bases (e.g. 0,1,2…) then the sources (e.g. 3, 4,5…) and eventually the customers.
Since we will have only one base, it always takes the location 0. Then you will have the *m* sources (from location 1 to m) and then the customers (from location m+1).

All Locations (Base, Sources and customers) common characteristics

- DISTMATRIX (*p, q*): distance between locations *p* and *q* (in km);
- TIMEMATRIX (*p, q*): travel time from location *p* to location *q* (in minutes).

Sources and customers characteristics

- SETUPTIME (*p*): the fixed part of loading/delivery time for point *p* (delivery for a customer or loading for a source) (in minutes)
- ALLOWEDTRAILERS (*p*): set of trailers allowed to supply the customer p or load from source *p*.

CUSTOMER characteristics:

- CALLIN(p): if 1, the customer is call-in, if 0, the customer is VMI
- TIMEWINDOWS (*p*): the set of opening hours intervals of customer *p*, each within the interval [0, T]. Deliveries should happen within those intervals.
- LAYOVERCUSTOMER(p): if 1 the customer must be served in a shift with a layover, 0 otherwise.

CALL-IN CUSTOMER characteristics

If CALLIN(P) = 1

- ORDERS(P) the set of orders we must respect, each of them have the following structure:
  - ORDERQUANTITY(O,P): the ordered quantity.
  - ORDERQUANTITYFLEXIBILITY (o,p): the minimum ratio]0%,100%] of the ordered quantity to deliver on customer p in order to consider the order o as satisfied.
  - EARLIESTTIME(o,p) = the earliest start (in minutes) for all the delivery operations related to the order o for the customer p. The value should be in the interval [0, T[
  - LATESTTIME(o,p) = the latest end (in minutes) for all the delivery operations related to the order o for the customer p. The value should be in the interval]EARLIESTTIME(o,p), T]

Important remark: Orders never overlay in time.

VMI CUSTOMER characteristics

If CALLIN(p) = 0

- SAFETYLEVEL (*p*): The level at the customer tank must always be above the Safety Level to avoid product shortage;
- *FORECAST(p,h) :* the amount of product in mass (Kg) that is used by the customer at location p during the timestep h;
- *INITIALTANKQUANTITY(p):* the amount of product in mass (Kg) available in the customers tank at the beginning of the horizon;
- CAPACITY(*p*): the maximal amount of product in mass (kg) that can be delivered to a customer at location p;
- MINIMUMOPERATIONQTY(p): the minimum amount of product that should be delivered in a single operation.

4. FORMULATION

4.1. Variables

A solution of the problem is a set of shifts (denoted by **SHIFTS**).
The following decision variables are defined on each shift *s*∈**SHIFTS**:

- *driver* (s): the driver for shift *s*
- *trailer* (s): the trailer (and associated tractor) for shift *s*
- *start (s)*: the starting time for the shift *s* (within [0, T[in minutes)
- *Operations(s)*: it is a list of operations (loadings, deliveries) performed during the shift

Each operation *o* ∈ **OPERATIONS(s)** is defined by:

- *arrival* (*o*): the arrival time of operation *o* (within [0, T[in minutes)
- *point* (*o*): the location at which operation *o* takes place either sources or customers
- *quantity* (*o*): the quantity to be delivered or loaded in operation o. It is negative for loading operations at sources, positive for delivery operations at customers. (in kg)

4.2. Constraints

Notes:

- *all the operators prev, final, start, end… used in this section are defined in the appendix.*
- *The labelling of the constraints is made as follows: unless explicitely specified the constraints have the same formulation as version 1. When the constraint is reformulated or added to be compliant with version 2, the constraint label will ends with _V2 . This labelling is consistent with the labelling available in the checker V2.*

4.2.1. Bounding constraints

All unary constraints on variables (e.g., non-negativity, inclusion in [0, T[, lower and upper bounds specified in previous section) must be satisfied.

4.2.2. Constraints related to layovers

[LAY01 V2] any travel between two locations that lasts more than layoverduration plus drivingtime are considered as layover

For a given shift s, for any operation o, including the final operation at the base let's define layoverafter(o) which indicates whenever there is a layover after the operation o.

For a given shift s ∈ Shifts, *for any operation o* ∈ { *Operations* (s) U *final*(s)}

If (arrival(o) – departure(prev(o)) >= LAYOVERDURATION*(driver(s))* + TIMEMATRIX*(point(prev(o)),point(o)) then*

layoverbefore(o) = 1

else

layoverbefore(o) = 0

For the *departure*(o) definition, please refer to constraint **SHI03**

[LAY02_V2] A shift must include a Layover if and only if there is one or more deliveries to layover customers

For a all shift s ∈ Shifts*,*

If there is no customer p ∈ Customers(s) where LAYOVERCUSTOMER(p) = 1 in customer(s) then ,

$$\sum_{\substack{o \in Operations(s) \\ \cup final(s)}} LayoverBefore(o) = 0$$

[LAY03_V2] Only one layover per shift is allowed.

For a all shift s ∈ Shifts*,*

$$\sum_{\substack{o \in Operations(s) \\ \cup final(s)}} LayoverBefore(o) \leq 1$$

4.2.3. Constraints related to drivers

[DRI01 | Inter-Shifts duration]

For each driver *d*, two consecutive shifts assigned to *d* must be separated by a duration of MININTERSHIFTDURATION (d).

For all d ∈ D*RIVERS For all* s_1, s_2 ∈ *shifts (d),*

Start (s_2) > *end* (s_1) + MININTERSHIFTDURATION (*d*) *OR start* (s_1) > *end* (s_2) + MININTERSHIFTDURATION (*d*)

[DRI03_V2 | Respect of maximal driving time]

For each operation associated with a shift *s (including the final(s) operation performed at the base)*, the cumulated driving time is the total travel time on the shift up to the previous operation plus the travel time from the location of the previous operation to the location of the current operation.

If the shift contains a layover, the cumulated driving time for each operation is counted from the beginning of the shift, in the case of the operation start before the layover, or from the layover, otherwise.

Please note that it is possible for a driver to drive in just before the layover, in order to satisfy the maximum driving duration constraint.
In order to take in account this eventuality, we will define an intermediate variable drivingtimebeforelayover(o), which is the minimum time driven by the driver just before taking the layover just before the operation o, in order to satisfy the maximum driving duration of the driver on the whole shift.

For a given operation o in a shift s where layoverbefore(o)=1 let's define

$$drivingTimebeforelayover(o) = Min\begin{pmatrix} MaxDrivingDuration(driver(s)) - cumulatedDrivingTime(prev(o)), \\ timeMatrix(prev(o), o) \end{pmatrix}$$

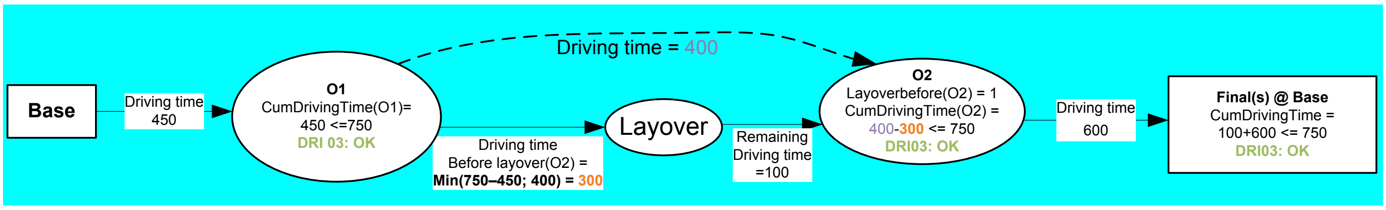
Using this definition, let's now define the cumulateddrivingtime(o) for each operation in the shift.

For a given s ∈ SHIFTS,
For all o ∈ {Operations(s) ∪ final(s)}
If layoverbefore(o) then
cumulatedDrivingTime(o) = timeMatrix(prev(o),o) - drivingTimebeforelayover(o)
else
cumulatedDrivingTime(o) = cumulatedDrivingTime(prev(o)) + timeMatrix(prev(o),o)

Then, the constraint formulation is: for each operation, the cumulated driving time of the shift cannot exceed (by law) the maximum allowed driving time

For all s ∈ SHIFTS, For all o ∈ {Operations(s) ∪ final(s)}
cumulatedDrivingTime(o) ≤ MaxDrivingDuration(driver(s))

Example, let's have shift s were maxDrivingDuration(driver(s)) is 750 minutes:



In this case, the driving time before layover is used to gain driving time in order to serve a layover customer in the operation O2. In this particular case, both customer(O1) and customer(O2) are layover customers.

Please note that all constraints DRI03 are valid in every operation including final(s).

[DRI08| Time windows of the drivers]

For each shift *s*, the interval [start(*s*), end(*s*)] must fit in one of the time-windows of the selected driver.

For all s ∈ SHIFTS
It exists at least a tw∈TIMEWINDOWS(Drivers(s)), start(s)≥start(tw) and end(tw)≥end(s)

4.2.4. Constraints related to trailers

[TL01 | Different shifts of the same trailer cannot overlap in time]

Consider any two shifts s_1 and s_2 that use the same trailer. Then, either s_1 ends before the start of s_2 or s_2 ends before the start of s_1.

For all tl ∈ TRAILERS For all $s_1, s_2 \in$ *shifts(tl),*
$start(s_2) > end(s_1)$ *or* $start(s_1) > end(s_2)$

[TL03_V2| The trailer attached to a driver in a shift must be compatible]

For each shift s, the assigned trailer must be one of the trailers that can be driven by the driver.

For all s ∈ SHIFTS, trailer(s) ∈TRAILERS(driver(s))

4.2.5. Constraints related to sites

[DYN01_V2 | Respect of tank capacity for each site (except call-in customers)]

For each site p, the tank quantity at each time step *h* must be contained into the interval [0, CAPACITY(P)].

For all p ∈ {CUSTOMERS | CALLIN(P)=0}, For all h ∈ [0, H[

$0 \leq tankQuantity(p, h) \leq$ CAPACITY(*p*)

For each customer *p*, the tank quantity at each time step *h* is equal to the tank quantity at time step *h - 1*, minus the forecasted consumption at *h*, plus all the deliveries performed at *h*.

Let us remind that the values of *Forecast(s,h)* and $\sum_{o \in Operations(p,h)} quantity(o)$ are positive for customers. This allows writing the same basic inventory dynamic equation for sources and customers.

For all *p* ∈ {CUSTOMERS | CALLIN(P)=0}

tankQuantity(p,-1) = *INITIALTANKQUANTITY(p)*

For all h ∈ [0, H[

Given that *dyn = tankQuantityl(p,h-1) – FORECAST(p,h) +* $\sum_{o \in Operations(p,h)} quantity(o)$

tankQuantity(p,h) = max(dyn,0)

Note: We assume that the entire quantity delivered in an operation is available in the customer tank as soon as the truck arrives at the customer (even if the trailer should stay a fixed time to complete the delivery, as explained in constraint SHI03).

4.2.6.Constraints related to shifts

[SHI02_V2 | Arrival at a point requires traveling time from previous point, and eventually the layover duration]

For o∈Operations(s), for all shift s,

*arrival(o) ≥ departure(prev(o)) + TIMEMATRIX(prev(o),o) + layoverbefore(o)*LayoverDuration(driver(s))*

Therefore, since each shift ends at the base, we need to take in account the travel from the last operation to the base (final(s)):

For all s ∈ SHIFTS
arrival(s) ≥ arrival(final(s))

Note: the waiting time of the drivers (at the gate of a source or a customer) is not defined explicitely in this model. Therefore, an arbitrary long "idle time" (where the driver "rests at the door of the customer or source doing nothing") can precede any operation belonging to a shift, as long as all of the constraints are respected. In particular, the constraint LAY01 forces to consider a layover if the idle time is greather than a maximum

[SHI03 | Loading and delivery operations take a constant time]

departure(o) = arrival(o) + SetupTime(point(o))

[SHI04_V2 | delivery operations are performed during opening hours of customers]

For all Operations, the interval [arrival(o), departure(o)] must be fully included in one of the opening time windows of the site.

For all s∈ *SHIFTS*

For all *o ∈ Operations(s) it exist a tw in TIMEWINDOWS(point(o)), such that*

arrival(o) ≥ start(tw) and end(tw) ≥ departure(o)

[SHI05 | delivery operations require the customer site to be accessible for the trailer]

Forall s∈SHIFTS,o∈Operations(s): if point(o) ∈ [CUSTOMERSUSOURCES] *then trailer(s)* ∈ ALLOWEDTRAILERS(point(o))

[SHI06 | trailerQuantity cannot be negative or exceed capacity of the trailer]

For a given shift s ∈ SHIFTS, For all o ∈ Operations(s) with {final(s}),
trailerQuantity(o) = trailerQuantity(prev(o)) - quantity(o)
trailerQuantity(o) ≥ 0
trailerQuantity(o) ≤ CAPACITY(trailer(s))

[SHI07 | Initial quantity of a trailer for a shift is the end quantity of the trailer following the previous shift.]

endTrailerQuantity(s) = trailerQuantity(last(Operations(s))

From these definitions we can derive this constraint:
if (s = first(shifts(trailer(s)))
startTrailerQuantity(s) = INITIALQUANTITY(trailer(s))
else
startTrailerQuantity(s) = endTrailerQuantity(prev(s, shifts(trailer(s)))
endif

[SHI11 | Some product must be loaded or delivered]

For each source, some product (a negative quantity) must be loaded and for each customer, and some product (a positive quantity) must be delivered.

For all *p* ∈ CUSTOMERS, For all h ∈ [0, H-1], For all o∈ Operations(p,h)
quantity(o) ≥ 0
For all *p* ∈ SOURCES, For all h ∈ [0, H-1], For all o∈ Operations(p, h)
quantity(o) < 0

[SHI16 V2| Capacity at the customer's site (except call-in customers)]

For each delivery operation o in a shift s, the delivered quantity of a VMI customer must be smaller than the customer tank capacity, and greather than the minimum operation quantity for that customer.

For all *p* ∈ {CUSTOMERS | CALLIN(P)=0},
For all h ∈ [0, H-1], For all o∈Operations(p,h)
quantity(o) <= CAPACITY(*p*)
quantity(o) >= MINIMUMOPERATIONQUANTITY(*p*)

4.2.7. Constraints related to quality of service

[QS01_V2 | call in customer order satisfaction by one operation at least]

For call-in customers, each order od should be satisfied by one or more operations. That operation should begin after the earliest time and before the latest time of the order.

For all *p* ∈ { *c*∈ CUSTOMERS| CALLIN(C)=1},
For all *od* ∈ {ORDERS(P)},

$$\text{orderQuantityFlexibility(od)} * quantity(od) \leq \sum_{\substack{\text{op} \in \{\text{operations(s)}| \\ \text{earliestTime(od)} \leq \text{arrival(op) and} \\ \textbf{arrival(op)} \leq \textbf{latestTime(od)}, s \, \square \, Shifts\}}} Quantity(op) \leq quantity(od)$$

Important remark: Orders never overlay in time.

[QS03_V2 | call in customer each operation must be related to an order]

Each operation on a call-in customer should be related to an order, meaning no operations are possible if there is any related order.

For all s ∈ Shifts, For all op ∈ {operation(s)},

IF POINT(OP) ∈ {P ∈ CUSTOMERS | CALLIN(P)=1} then

It exists at least an order od ∈ ORDERS(POINT(OP)) such as arrival(op) ≥ earliestTime(od) and latestTime(od)≥ arrival (op)

Important remark: Orders never overlay in time.

[QS02_V2 | Run-out avoidance]

For each VMI customer *p*, the tank level must be maintained at a level greater than or equal to the safety level *SafetyLevel*(*p*), at all times.

For all *p* ∈ CUSTOMERS | CALLIN(P)=0 , For all h ∈ [0, H-1]

SafetyLevel(*p*) ≤ *tankQuantity(p, h)*

4.3. Optimization goal

4.3.1. Objective function

The goal is to minimize the distribution costs required to meet customer demands for product over the long term horizon. In order to tend to that final goal, we will minimize the **logistic ratio**.

The logistic ratio is defined as the total cost of the shifts divided by the total quantity delivered on those shifts:

$$LR = \frac{\sum_{s \in SHIFTS} Cost(s)}{TotalQuantity}$$

The cost of a shift represents the **distribution costs** related to the shift, including:

- the **distance cost,** applied to the total length of the shift, which is generally related to the trailer used (covering fuel consumption and maintenance)
- the **time cost** applies to the total duration of the shift, which is generally related to the driver (covering the driver salary and charges)
- the **layover cost**, if the shift contains a layover

$$Cost(s) = \begin{pmatrix} DISTANCECOST(trailer(s)) \times travelDist(s) + \\ TIMECOST(driver(s)) \times workingTime(s) + \\ LayoverCost(s) \end{pmatrix}$$

Where:

$$workingTime(s) = (end(s) - start(s)) - \left(\sum_{o \in OPERATIONS(s) \cup final(s)} LayoverBefore(o) * LayoverDuration(driver(s)) \right)$$

$$travelDist(s) = \sum_{o \in OPERATIONS(s) \cup final(s)} DISTMATRIX(\mathrm{point}(prev(o)), \mathrm{point}(o))$$

$$LayoverCost(s) = \sum_{o \in OPERATIONS(s) \cup final(s)} LayoverBefore(o) * LayoverCost(driver(s))$$

The total quantity delivered over all shifts is computed as follows:

$$TotalQuantity = \sum_{s \in SHIFTS} \sum_{\substack{o \in Operations(s) \\ quantity(o) > 0}} quantity(o)$$

Note that if T*otalQuantity = 0,* we will consider also *LR = 0.*

4.3.2. Time integrations over scheduling optimization horizons

The objective of an IRP is to minimize distribution costs over an enough long period (horizon) covering one or more replenishment cycles for all the customers.

If the time horizon is too short, the optimization can be short-sighted because of the “end-of-period side effect”: customers that do not strictly require delivery within this short horizon are excluded from the planning, which might increase the risk of shortage just beyond the horizon.

Considering a longer time horizon makes the relative impact of this side effect more and more negligible, but complexifies the problem and requires longer forecast on the customers, which could be far from the reality.

By consequence, a “good” value for the time horizon cannot be generalized as it depends on many factors, particularly the confidence in the customer forecast at various points over that horizon. For this reason, in this challenge, we ask the contenders to try different horizons, even for the same dataset.

4.4. APPENDIX : Specific Functions definition used in the constraint modeling

For the sake of readability, we use the following notations or intermediate variables.

- *shifts (d):= {s ∈ Shifts* such that *driver(s) = d},* i.e., the list of shifts that use driver d.
- *shifts (tl) := {s ∈ Shifts* such that *trailer(s) = tl},* i.e., the list of shifts that use trailer tl.
- *operations (s) = the union of all operations of the shift, including initial the initial operation.*
- *operations (tl) = the union of all operations of each trailer tl sorted by time.*
- *customers(s) = the union of all customers delivered on operations(s)*
- *customer(o) = the customer served in a operation o*

3 generic functions are defined and used to navigate within any non empty lists:

- *first* (*list*) returns the first item in the list
- *last* (*list*) returns the last item of the list.
- *prev (e, list)* returning the previous item *e* in list ***list*** (or e if e is the first item of the list)

Several customized functions can be used:

- *prev(o):* For each operation *o* we define *prev(o)* as the predecessor of *o* in the list of operations of the prev(o):=shift(prev(o, Operations(shift(s)))). If *o* is the first operation of the shift then we extend this definition by defining *prev(o)* as a "virtual" first operation *f* with the following characteristics
    - *shift(f) := shift(o)*
    - *point(f) := Base(shift(o))*
    - *trailerQuantity(f)=startTrailerQuantity(shift(o))*
    - *quantity(f) := 0*
    - *arrival(f) := start(shift(o))*
    - *cumulatedDrivingTime(f)* := 0
    - *departure(f) := departure(shift(o))*

- *final (s):* a final operation is created at the end of the shift with the same attributes as *operations* defined above but with null volume. This last stop is only defined in order to simplify the subsequent writing of constraints by making the number of operations equal to the number of arcs (travel from one point to another) in the shift. It hosts variables related to the last travel from the last operation point to the base. The following attributes of this operation have constrained values:
    - *point (final (s)) = base (s)*
    - *quantity(final (s)) = 0*
    - *arrival (final (s)) = arrival (s)*
    - departure (final (s)) = end (s)

Please note that the layoverbefore(final(s)) is NOT constrained, and must be 1 if there is a layover before returning to the base, and 0 otherwise.

- For any (delivery or load) point *p* and timestep *h*, the set of all Operations at point *p* taking place during timestep *h* is defined by:

  Operations(*p* ,*h*) := { *o* ∈ Operations such that *point(o)* = *p* and rounddown(*arrival(o)* / U) = h}

Finally the start and end of a time window *tw* are referred to as *start* (tw) and *end* (tw) respectively.