BACHELOR'S DEGREE THESIS

Degree in Mathematics

Title: Mathematical Programming applied to Cost to Serve Optimization

Author: Santiago Llaquet Vélez

Advisor: Francisco Javier Heredia Cervera

Department: Department of Statistics and Operations Research

Academic year: 2022-2023

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Degree in Mathematics

Bachelor's Degree Thesis

Mathematical Programming applied to Cost to Serve Optimization

Santiago Llaquet Vélez

Supervised by
Andrés Puy - Data Science Consultant at Accenture

January 2023

Thanks to Andrés for all that I have learnt from him, for all the time invested in teaching me and helping me in this project, and for always looking at the bright side of things. Thanks to Sandra for her guidance and support, for always trying to help and be involved, and for the opportunity to work on this thesis as an Innovation project at Accenture. Thanks to Javier for all the suggestions and the expertise in Operations Research. Thanks to Ade, to my family and to my friends for all the support that they have given me and for always being interested in the project.

Abstract

The Cost to Serve problem is a logistics problem, mainly in the Supply Chain field, which uses advanced analytics and often has a strategic point of view. Its main goal is to minimize logistic costs and, at times, its focus might shift towards environment care. Moreover, it also reveals the real cost of serving individual customers, allowing re-segmentation of the enterprise's customers based on customer's profitability.

The logistic costs are usually split into warehousing costs, picking costs and transportation costs. However, we will only be considering transportation costs in this project. We will also be using a general time-view (it could be annually, on a quarterly basis, monthly...) this way we can have a better global understanding of the data while not having too broad of a view on it. The drawback of this general view is that we cannot differentiate between different time instants: since we work with totals and averages we are supposing that the data is homogeneously distributed over the time-period considered.

In this thesis we consider a version of the Cost to Serve problem, give a MILP formulation to solve it, and study how small perturbations of the parameters affect it.

Keywords

Cost to Serve, Mixed-Integer Linear Programming

Contents

1. Introduction

1.1 Real-world problem

Trade and commerce, originated from communication on prehistoric times, have played a crucial role on the development of humanity ever since. In the last decades, freight transportation has become very important, mainly due to its integration in Supply Chain caused by the increase of commerce at a global scale. As a result, freight transport is now the backbone of the industrial system, whith great implications in the world economy.

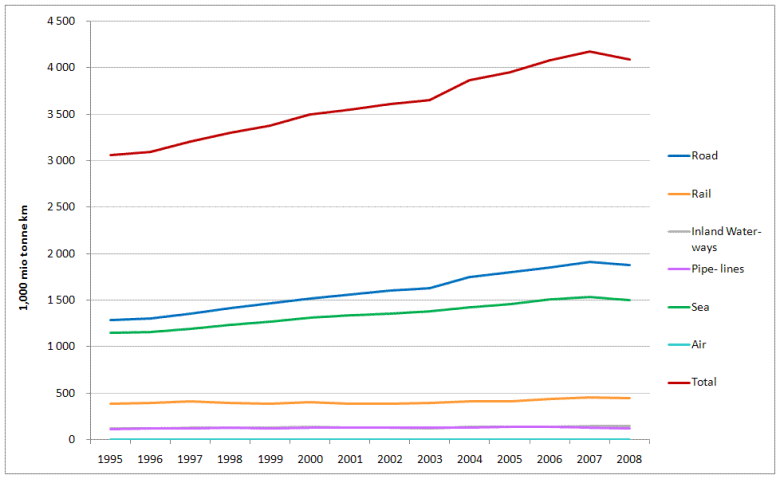


Figure 1: Freight transport demand by mode - EU27 [1]

This increase in freight transportation has also signified an increment of the CO_2 emissions produced by those transportations.

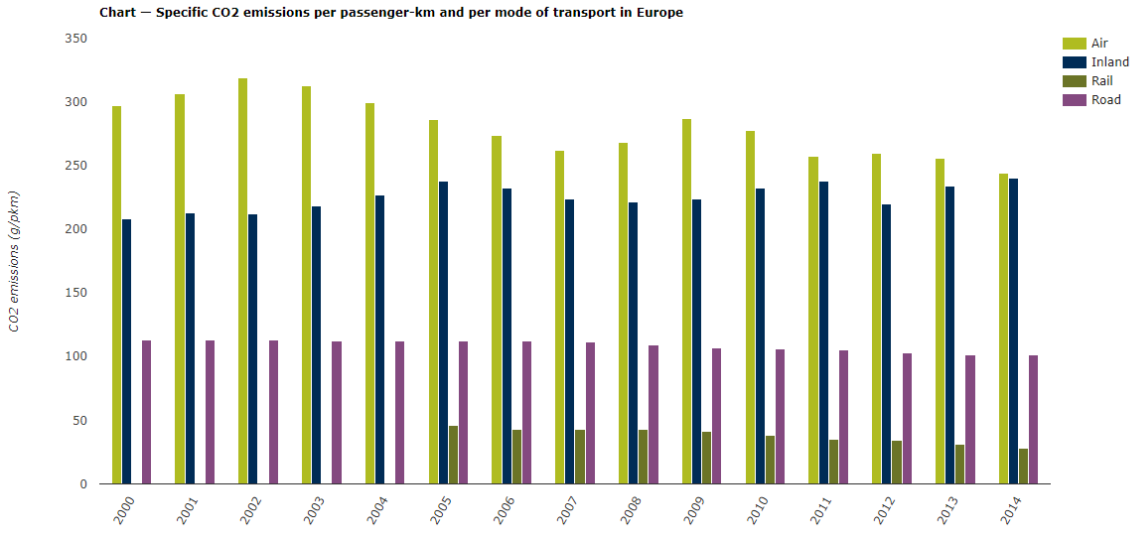


Figure 2: Specific CO_2 emissions per transport over the years in Europe [3]

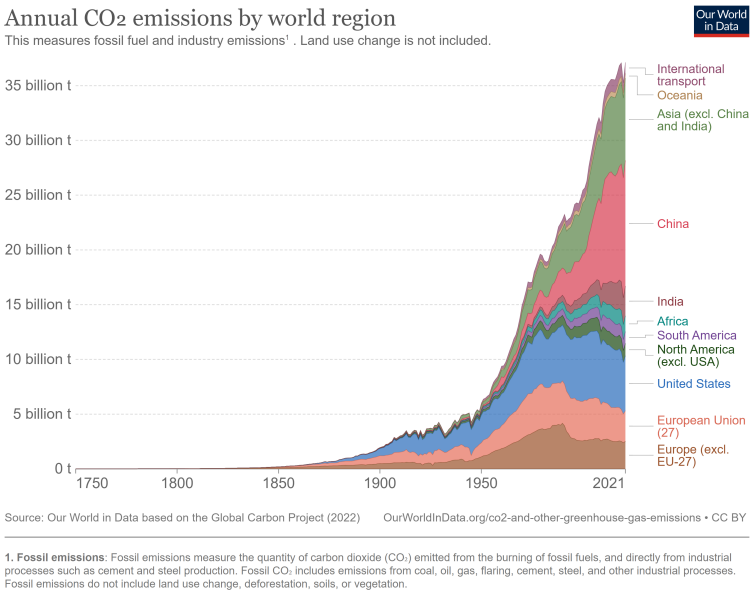


Figure 3: Annual CO_2 emissions over the years [2]

This arises a problem for transportation companies: to both reduce their transportation costs and the environmental damages produced by those transportations.

1.2 Mathematical problem

The Cost to Serve problem is the problem of minimizing the costs produced by freight transportation, and its goal is also to reveal the real cost of serving individual customers to allow their re-segmentation based on their profitability. Thus, the objective of this thesis is to minimize the freight transportation costs while finding out customer's profitability.

In this problem, for a particular time-period, we consider a set of ship-tos (they will also be referred to as destinations or shops) where each one demands to be served a certain amount of some commodities (measured in pallets) in the time-period, and a set of ship-froms (from now on they will also be referred to as origins or warehouses) where each one has a certain quantity of products stored during the time-period. We also consider different transport means —with different characteristics— that can transport the products from the origins to the destinations.

To minimize the costs and optimize the shipments we will use Mathematical Programming. The direct formulation of the problem is a quadratic one. However, some Mathematical Optimization solvers can not deal with quadratic problems because they are only able to optimize linear ones. Other, more modern, solvers do accept quadratic problems, but they tend to converge in a slower way than linear ones. For these reasons quadratic problems are frequently linearized. The goal of this thesis is not to just build a model that solves the Cost to Serve problem, but to build one that can solve it with real-world instances of it (i.e. with big networks). Therefore, the model will have to give out a solution that can scale rapidly as the problem's instances increase in size. Consequently, we will linearize the quadratic parts of our problem.

Another approach to solve the Cost to Serve problem would be to separate the problem into two subproblems: the first one would focus on determining the ways that shipments could be made (something that will be called multi-drop and will be explained later on), and the second one would try to resolve the

freight flux. That is, another way to solve the Cost to Serve problem would be to divide it into a clustering problem and a flux problem.

1.3 State of the art

Little research has been done on Cost to Serve Optimization: an explanation on what the Cost to Serve is can be found in [5], a model that solves the Cost to Serve problem dividing the problem into subproblems and using Machine Learning techniques can be found in [8], and a model which uses a Genetic Algorithm to solve a somewhat similar version of the Cost to Serve problem that we are considering (although with less features and some different considerations) can be found in [6]. Anyway, we could not find any literature on other similar versions of the Cost to Serve problem that we are considering.

2. Parameters and variables

A descriptive list of the parameters and variables used in the formulation will follow. Note that during the whole thesis the variables will be written in bold, while the parameters will not. We will also choose I, J, K and P to be respectively the sets of origins, destinations, transport means, and products.

2.1 Parameters

- ν: number of days in the time-period.
- $\bar{\nu}$: number of weeks in the time-period.
- θ_k: coefficient that translates a kilogram of emissions of the transport $k \in K$ into monetary units.
- M: big enough positive constant.
- $z_{j,p}$: amount of pallets requested by the ship-to $j \in J$ of the product $p \in P$.
- σ_p: weight of a pallet of the product $p \in P$.
- μ_p: sell price of a product $p \in P$.
- $\bar{\mu}_p$: acquisition cost of a product $p \in P$.
- $N_{i,p}$: amount of pallets of the product $p \in P$ that the ship-from $i \in I$ has in the time-period.
- $\delta_{i,j,k}$: binary value which is 1 if the transport mean $k \in K$ can be used to send commodities between $i \in I$ and $j \in J$, and 0 otherwise.
- $\bar{N}_i$: maximum number of shipments that a warehouse $i \in I$ can do in the time-period.
- fc_k: fixed costs of transporting commodities in a vehicle of the type $k \in K$.
- vc_k: variable costs of transporting commodities in a vehicle of the type $k \in K$.
- ρ_k: fuel cost of a vehicle of the type $k \in K$.
- $\bar{\rho}_k$: fuel consumption of a vehicle of the type $k \in K$.
- φ_k: minimum cost of transporting some commodities in a vehicle of the type $k \in K$.
- $\bar{\varphi}_k$: minimum fill percentage where the shipping rate starts to apply for a transport mean of the type $k \in K$.
- Ω_k: tax per pallet in the vehicle $k \in K$ when the pallet fill rate is between φ_k and ψ_k.
- ψ_k: maximum cost of transporting some commodities in a vehicle of the type $k \in K$.
- $\bar{\psi}_k$: maximum fill percentage where the shipping rate does not apply anymore for a transport mean of the type $k \in K$.
- P_k: maximum amount of pallets admitted by a vehicle of the type $k \in K$.

- W_k: maximum weight load admitted by a vehicle of the type $k \in K$.
- R_k: amount of available vehicles of the transport mean $k \in K$ that can be used to do the shipments.
- h_k: kg of CO_2 per km emited by a transport mean of the type $k \in K$.
- $\kappa_{i,j,k}$: number of days taken to do a shipment from $i \in I$ to $j \in J$ with the transport mean $k \in K$.
- $\gamma_{i,j,k}$: distance between the ship-from $i \in I$ and ship-to $j \in J$ when using the transport mean $k \in K$.
- $\bar{\gamma}_{j_1,j_2,k}$: distance between the ship-tos $j_1, j_2 \in J$ when using the transport mean $k \in K$.
- $\bar{\tau}_j$: contractual delivery frequency of $j \in J$ stipulated.
- Φ_i: number of centers of clusters, or number of clusters, to cluster the ship-tos from the perspective of the ship-from $i \in I$.
- Σ_k: desired profitability on the shipments with the transport mean $k \in K$.

2.2 Variables

- $\mathbf{c_{i,j}}$: binary variable with value 1 if $j \in J$ is the center of a cluster for the groups in $i \in I$, and 0 otherwise.
- $\mathbf{b_{i,j_1,j_2}}$: binary variable with value 1 if the ship-to $j_2 \in J$ is assigned to the cluster centered at $j_1 \in J$ of the groups for $i \in I$, and 0 otherwise.
- $\mathbf{tc}$: monetary cost of the shipments due to direct transportation costs.
- $\mathbf{ec}$: monetary costs due to the environmental costs of the shipments.
- $\mathbf{sc_{i,j,k}}$: shipping costs of the shipments from $i \in I$ to $j \in J$ due to the shipping fees of transporting a certain load of pallets in a transport mean $k \in K$.
- $\mathbf{lc_{i,j,k}}$: cost of the shipments from $i \in I$ to $j \in J$ with $k \in K$ due to the fixed, variable, and fuel consumption costs.
- $\mathbf{s_{i,j,k}}$: number of shipments done between $i \in I$ and $j \in J$ with the transport mean $k \in K$.
- $\bar{\omega}_{\mathbf{i,j_1,j_2,k}}$: dummy variable used for linearizing the logistic costs definition constraint. It has a negative value if $j_2 \in J$ does not belong to the group centered at $j_1 \in J$ for $i \in I$, and its value is $\mathbf{s_{i,j_1,k}}$ if it does belong to it.
- $\omega_{\mathbf{i,j_1,j_2,k}}$: variable used to linearize the logistic costs definition constraint. Its value is 0 if $\bar{\omega}_{\mathbf{i,j_1,j_2,k}}$ is negative and it is $\bar{\omega}_{\mathbf{i,j_1,j_2,k}}$ if it is positive.
- $\bar{\alpha}_{\mathbf{i,j,k}}$: dummy variable with real values used to decide if the shipments from $i \in I$ to $j \in J$ with $k \in K$ pay the minimum fee or not.
- $\hat{\alpha}_{\mathbf{i,j,k}}$: positive variable used to decide if the shipments from $i \in I$ to $j \in J$ with $k \in K$ pay the minimum fee or not.

- $\alpha_{i,j,k}$: binary variable with value 1 if the shipments from $i \in I$ to $j \in J$ with $k \in K$ pay the minimum fee due to not filling the transport mean k with less pallets than the minimum fill rate, and 0 otherwise.
- $\bar{\beta}_{i,j,k}$: dummy variable with real values used to decide if the shipments from $i \in I$ to $j \in J$ with $k \in K$ pay the maximum fee or not.
- $\hat{\beta}_{i,j,k}$: positive dummy variable used to decide if the shipments from $i \in I$ to $j \in J$ with $k \in K$ pay the maximum fee or not.
- $\beta_{i,j,k}$: binary variable with value 1 if the shipments from $i \in I$ to $j \in J$ with $k \in K$ pay the maximum fee due to filling the transport mean k with more pallets than the maximum fill rate, and 0 otherwise.
- $\bar{\xi}_{i,j,k}$: dummy variable with real values to compute $\xi_{i,j,k}$.
- $\xi_{i,j,k}$: shipping costs of transporting commodities from $i \in I$ to $j \in J$ with $k \in K$ if the minimum fee applies.
- $\bar{\eta}_{i,j,k}$: dummy variable with real values used to compute $\eta_{i,j,k}s$.
- $\eta_{i,j,k}$: shipping costs of transporting commodities from $i \in I$ to $j \in J$ with $k \in K$ if the minimum and maximum fees do not apply.
- $\bar{\lambda}_{i,j,k}$: dummy variable with real values used to compute $\lambda_{i,j,k}$.
- $\lambda_{i,j,k}$: shipping costs from transporting commodities from $i \in I$ to $j \in J$ with $k \in K$ if the maximum fee applies.
- $\bar{p}_{i,j_1,j_2,p,k}$: dummy variable used to compute the amount of pallets of the product $p \in P$ transported with $k \in K$ from $i \in I$ to the cluster centered at $j_1 \in J$, with travel destination $j_2 \in J$.
- $p_{i,j_1,j_2,p,k}$: variable that shows the amount of pallets of the product $p \in P$ transported with $k \in K$ from $i \in I$ to the cluster centered at $j_1 \in J$, with travel destination $j_2 \in J$.
- $\bar{w}_{i,j_1,j_2,p,k}$: dummy variable used to compute the weight of the pallets of the product $p \in P$ transported with $k \in K$ from $i \in I$ to the cluster centered at $j_1 \in J$, with travel destination $j_2 \in J$.
- $w_{i,j_1,j_2,p,k}$: variable that shows the weight of the pallets of the product $p \in P$ transported with $k \in K$ from $i \in I$ to the cluster centered at $j_1 \in J$, with travel destination $j_2 \in J$.
- $y_{i,j,p}$: number of pallets that are sent from $i \in I$ to $j \in J$ of the product $p \in P$.
- $x_{i,j,p,k}$: number of pallets of the product $p \in P$ sent from $i \in I$ to $j \in J$ with the transport mean $k \in K$.
- τ_j: weekly deliveries to the ship-to $j \in J$.

3. Mathematical formulation

3.1 Considerations for solving the problem

A naive way to solve the problem would be to simply fill vehicles on the warehouses with each vehicle having just one shop as a destination, and when the vehicles were filled enough they would be sent. However, it could happen that two destinations were close enough such that the shipments that were sent from an origin to them would be optimized if the origin sent to both shops together, that is, if the vehicles were filled with commodities requested by both destinations.

For example, consider a warehouse and two shops located at A, B and C respectively, where the distances between the warehouse and the shops are way larger than the distance between the shops. Consider also that both shops request to be served half a truck-load. While the cost of the naive solution would be the cost of sending two trucks, by merging the shipments we could get a cost of sending just one truck. Thus, by having the possibility of merging shipments we might be able to drastically reduce the total costs. This technique, where a shipper performs multiple drop-offs along a specified route instead of completing the entire shipment on just one drop, is called multi-drop.

In our problem, instead of having the possibility of combining particular shipments, since we are using a general time-view to solve the Cost to Serve problem, we will consider the possibility of clustering shops into groups. By that we mean that each ship-from will have different clusters, each formed by some shops (we will impose that all shops must belong to one group). That way, for each ship-from, the commodities to all shops belonging to a cluster will be put together so that the number of shipments can be reduced.

3.2 Objective function

The goal of this thesis is to minimize the total costs of transporting some commodities to some locations. It is hard to define what the total real cost is so, in our case, we will only be considering the costs directly caused by the movement of the sent vehicles that transport commodities from ship-froms to ship-tos. Then, these total costs can be split into transportation costs and environmental (or emission) costs.

We can consider two hyper-parameters: the weights c_1 and c_2, to be able to assess different relevance to the transportation and emission costs. Therefore, the function that has to be minimized is $c_1 \cdot \mathbf{tc} + c_2 \cdot \mathbf{ec}$, i.e. our goal will be:

$$\min c_1 \cdot \mathbf{tc} + c_2 \cdot \mathbf{ec}$$

3.3 Cluster constraints

We have already stated that for every ship-from we want to group the ship-tos so that the number of shipments can be reduced. To use this multi-drop technique, we can represent every group by a ship-to, which will be the center of the group (it could also be seen as the capital of the group), and the shops that belong to the group will be assigned to the group's center.

The constraints that have to be set to build the clusters will be:

$$\mathbf{b_{i,j,j}} = \mathbf{c_{i,j}},\ \forall i \in I,\ \forall j \in J \tag{1}$$

$$\mathbf{b_{i,j_1,j_2}} \leq \mathbf{c_{i,j_1}},\ \forall i \in I,\ \forall j_1, j_2 \in J \tag{2}$$

$$\sum_{j_1 \in J} \mathbf{b_{i,j_1,j_2}} = 1,\ \forall i \in I,\ \forall j_2 \in J \tag{3}$$

$$\sum_{j \in J} \mathbf{c_{i,j}} = \Phi_i,\ \forall i \in I \tag{4}$$

Constraint (1) means that, for a warehouse i, a shop j is assigned to the group centered in itself if and only if it is a center of the group clusters of i.

Constraint (2) means that a shop j_2 can be assigned to the cluster centered at j_1 for the warehouse i if j_1 is indeed a center of a group of i.

Constraint (3) ensures that all ship-tos are assigned to one, and only one, group for each warehouse.

Constraint (4) specifies the number of clusters that have to be formed for each warehouse.

3.4 Cost definitions

We subdivided the total costs into transportation and emission costs. In the next constraint we subdivide the transportation costs into costs generated by the transportation of commodities (referred to as logistic costs) and the ones caused by the shipping variable fees due to filling rates (referred to as shipping costs).

$$\mathbf{tc} = \sum_{i \in I, j \in J, k \in K} \mathbf{sc_{i,j,k}} + \mathbf{lc_{i,j,k}} \tag{5}$$

We will now focus on defining these costs. Defining the routes that the vehicles travel on is outside of our scope so, instead of computing the best routes for the shipments computing a hamiltonian path for each created cluster (it is an NP-complete problem and thus difficult to solve and computationally expensive), we define a simpler route for the clusters that the vehicles have to follow: for each shipment that is sent to a certain group, the vehicle that transports it travels first to the center and then, for each shop in the group, it travels to it and goes back to the center. That is, the vehicles have to travel twice the sum of the distances of the center of the cluster to the rest of the shops of the group plus the distance from the warehouse that they come from to the center of the group. The great advantage of this definition of distance travelled is that it is very easy to compute.

Having defined the route distances calculation, we can define the logistic costs:

$$\mathbf{lc_{i,j_1,k}} = \mathbf{s_{i,j_1,k}} \cdot \left(fc_k + [vc_k + \rho_k \cdot \bar{\rho}_k] \cdot \left[\gamma_{i,j_1,k} + 2 \cdot \sum_{j_2 \in J} \bar{\gamma}_{j_1,j_2,k} \cdot \mathbf{b}_{i,j_1,j_2} \right] \right),\ \forall i \in I,\ \forall j_1 \in J,\ \forall k \in K$$

The logistic costs of a shipment from i to j_1 with transport mean k are due to: the fixed, variable, and fuel consumption costs; and they also depend on the distance travelled, which has been defined as the distance from i to j_1 with the vehicle k plus twice the distance from j_1 to the rest of the shops in the

group centered in j_1 with the vehicle k. We observe that if j_1 is not a center of a group cluster of i, then, by constraint (2), $\mathbf{b}_{i,j_1,j_2} = 0\ \forall j_2 \in J$.

The right hand side of the equation is defined as a quadratic function (it actually is a quadratic term plus a linear one) and, for the reasons explained before, we want to linearize the quadratic part. To do that, we can create a dummy variable that tells us what the product $\mathbf{s_{i,j_1,k}} \cdot \mathbf{b_{i,j_1,j_2}}$ is. We will use max and min constraints in the formulation in order to reduce the notation and simplify the reading. However, they have to be replaced to obtain a linear programming problem. This will be discussed in section 4.1.

We first define a dummy variable $\bar{\omega}_{\mathbf{i,j_1,j_2,k}}$ whose value is the number of shipments made between i and j_1 with k if j_2 belongs to the group centered at j_1 of the centers of i, and its value is negative otherwise. Note that here we need a big enough constant M. After that, the value of $\mathbf{s_{i,j_1,k}} \cdot \mathbf{b_{i,j_1,j_2}}$ will just be the maximum between 0 and $\bar{\omega}_{\mathbf{i,j_1,j_2,k}}$.

$$\bar{\omega}_{\mathbf{i,j_1,j_2,k}} = \mathbf{s_{i,j_1,k}} - M \cdot (1 - \mathbf{b}_{i,j_1,j_2}),\ \forall i \in I,\ \forall j_1, j_2 \in J,\ \forall k \in K \tag{6}$$

$$\omega_{i,j_1,j_2,k} = \max(0, \bar{\omega}_{i,j_1,j_2,k}),\ \forall i \in I,\ \forall j_1, j_2 \in J,\ \forall k \in K \tag{7}$$

Then, the constraint that we wanted can be rewritten as:

$$\mathbf{lc}_{i,j_1,k} = \mathbf{s_{i,j_1,k}} \cdot fc_k + (vc_k + \rho_k \cdot \bar{\rho}_k) \cdot \left(\gamma_{i,j_1,k} \cdot \mathbf{s_{i,j_1,k}} + 2 \cdot \sum_{j_2 \in J} \bar{\gamma}_{j_1,j_2,k} \cdot \boldsymbol{\omega}_{\mathbf{i,j_1,j_2,k}} \right),\ \forall i \in I,\ \forall j_1 \in J,\ \forall k \in K \tag{8}$$

For some vehicles, on top of the fixed costs there are some shipping fees instead of the variable costs. These shipping fees only apply when the pallet fill rate of a shipment is in a certain region contained in $[0, 1]$ (the fill rate is measured as the total number of pallets sent in the vehicle over the total number of pallets that can be fitted in the vehicle). If the pallet fill rate is below (respectively over) a certain threshold, then a minimum (respectively maximum) fixed cost has to be payed. When the fill rate is between the minimum and maximum thresholds, the amount payed is the number of pallets sent in the vehicle multiplied by a certain shipping rate.

With that, the shipping costs $\mathbf{sc_{i,j,k}}$, which are the sum of all the shipping costs of the shipments from i to j with k, are defined as:

$$\mathbf{sc_{i,j,k}} = \left\{ \begin{array}{c} \mathbf{s_{i,j,k}} \cdot \varphi_k,\ \text{if } \sum_{p \in P} \mathbf{x_{i,j,p,k}} \leq P_k \cdot \bar{\varphi}_k \cdot \mathbf{s_{i,j,k}} \\ \Omega_k \cdot \sum_{p \in P} \mathbf{x_{i,j,p,k}},\ \text{otherwise} \\ \mathbf{s_{i,j,k}} \cdot \psi_k,\ \text{if } \sum_{p \in P} \mathbf{x_{i,j,p,k}} \geq P_k \cdot \bar{\psi}_k \cdot \mathbf{s_{i,j,k}} \end{array} \right\}$$

We can define this peacewise function in a single line and in a continuous way by adding constraints (9) to (14), and then we can linearize it by adding constraints (15) to (21).

We will be using dummy variables that will tell us whether each case of the piecewise function is met or not.

$$\bar{\alpha}_{\mathbf{i,j,k}} = \bar{\varphi}_k \cdot P_k \cdot \mathbf{s_{i,j,k}} - \sum_{p \in P} \mathbf{x_{i,j,p,k}},\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{9}$$

The first term on the right hand side is just the maximum number of pallets that have to be sent from i to j with k in $\mathbf{s_{i,j,k}}$ shipments so that the minimum fee always applies, and the right hand side is the actual number of pallets that are sent. It should be reminded that we are working with totals and averages and not with particular shipments.

The value $\bar{\alpha}_{i,j,k}$ is negative if the shipments from i to j with k are more filled on average than the minimum fill rate of k, and it is positive if they are below the minimum fill rate. If it has negative value the minimum shipping fee does not apply, meanwhile, if it is positive it does.

$$\hat{\alpha}_{i,j,k} = \max(0, \bar{\alpha}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{10}$$

$$\alpha_{i,j,k} = \min(1, \hat{\alpha}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{11}$$

This way, $\alpha_{i,j,k} = 0$ if $\bar{\alpha}_{i,j,k}$ is negative, and $\alpha_{i,j,k} = 1$ if $\bar{\alpha}_{i,j,k}$ is positive. Since $\bar{\alpha}_{i,j,k} \in \mathbb{Z}$ because the two terms that define it must be integers, $\alpha_{i,j,k}$ will be a boolean variable, and its value will be 1 if the first case of the piecewise function is met, and 0 otherwise.

We now do exactly the same for the third case of the piecewise function:

$$\bar{\beta}_{i,j,k} = \sum_{p \in P} x_{i,j,p,k} - \bar{\psi}_k \cdot P_k \cdot s_{i,j,k},\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{12}$$

$$\hat{\beta}_{i,j,k} = \max(0, \bar{\beta}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{13}$$

$$\beta_{i,j,k} = \min(1, \hat{\beta}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{14}$$

As discussed for $\alpha_{i,j,k}$, $\beta_{i,j,k}$ is a boolean with value 1 if the third case holds and 0 otherwise.

Getting these constraints together we can define the shipping costs without a piecewise function:

$$sc_{i,j,k} = [s_{i,j,k} \cdot \varphi_k \cdot \alpha_{i,j,k}] + \left[\Omega_k \cdot \sum_{p \in P} x_{i,j,p,k} \cdot (1 - \alpha_{i,j,k} - \beta_{i,j,k})\right] + [s_{i,j,k} \cdot \psi_k \cdot \beta_{i,j,k}],$$
$$\forall i \in I,\ \forall j \in J,\ \forall k \in K$$

However, they have now been defined in a quadratic way, so we linearize them by doing:

$$sc_{i,j,k} = \max(0,\ s_{i,j,k} \cdot \varphi_k - M \cdot (1 - \alpha_{i,j,k})) + \max\left(0,\ \Omega_k \cdot \sum_{p \in P} x_{i,j,p,k} - M \cdot (\alpha_{i,j,k} + \beta_{i,j,k})\right) +$$
$$+ \max(0,\ s_{i,j,k} \cdot \psi_k - M \cdot (1 - \beta_{i,j,k})),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K$$

Note that here we need M to be big enough.

We now proceed to split the terms in this definition.

$$\bar{\xi}_{i,j,k} = s_{i,j,k} \cdot \varphi_k - M \cdot (1 - \alpha_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{15}$$

$$\xi_{i,j,k} = \max(0,\ \bar{\xi}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{16}$$

$$\bar{\eta}_{i,j,k} = \Omega_k \cdot \sum_{p \in P} x_{i,j,p,k} - M \cdot (\alpha_{i,j,k} + \beta_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{17}$$

$$\eta_{i,j,k} = \max(0,\ \bar{\eta}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{18}$$

$$\bar{\lambda}_{i,j,k} = s_{i,j,k} \cdot \psi_k - M \cdot (1 - \beta_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{19}$$

$$\lambda_{i,j,k} = \max(0,\ \bar{\lambda}_{i,j,k}),\ \forall i \in I,\ \forall j \in J,\ \forall k \in K \tag{20}$$

Observe that for every trio i, j, k only one of $\boldsymbol{\xi_{i,j,k}}, \boldsymbol{\eta_{i,j,k}}$ and $\boldsymbol{\lambda_{i,j,k}}$ is not null.

Finally:

$$\mathbf{sc_{i,j,k}} = \boldsymbol{\xi_{i,j,k}} + \boldsymbol{\eta_{i,j,k}} + \boldsymbol{\lambda_{i,j,k}}, \ \forall i \in I, \ \forall j \in J, \ \forall k \in K \tag{21}$$

This way we have defined in a linear way the shipping costs of the shipments from i to j with transport mean k.

We define the emission costs as:

$$\mathbf{ec} = \sum_{i \in I, j_1 \in J, k \in K} \theta \cdot \mathbf{s_{i,j_1,k}} \cdot h_k \cdot \left[\gamma_{i,j_1,k} + 2 \cdot \sum_{j_2 \in J} \bar{\gamma}_{j_1,j_2,k} \cdot \mathbf{b_{i,j_1,j_2}} \right]$$

And we linearize the constraint by using the variable $\boldsymbol{\omega_{i,j_1,j_2,k}}$, which was defined in (7):

$$\mathbf{ec} = \sum_{i \in I, j_1 \in J, k \in K} \theta \cdot h_k \cdot (\mathbf{s_{i,j_1,k}} \cdot \gamma_{i,j_1,k} + 2 \cdot \sum_{j_2 \in J} \bar{\gamma}_{j_1,j_2,k} \cdot \boldsymbol{\omega_{i,j_1,j_2,k}}) \tag{22}$$

3.5 Assignment of the orders

Since we know the orders, i.e. the amount of pallets $\mathbf{z_{j,p}}$ of each product $p \in P$ that each ship-to $j \in J$ requests, our goal is to best distribute these orders among the warehouses to minimize the total cost.

Given the amount of pallets that the ship-to $j \in J$ requests of the product $p \in P$, $\mathbf{y_{i,j,p}}$ are the amount of pallets that we assign to the ship-from $i \in I$ of the product p that j ordered. Adding the contributions of all warehouses we impose that all the ordered products are assigned:

$$\sum_{i \in I} \mathbf{y_{i,j,p}} = \mathbf{z_{j,p}}, \ \forall j \in J, \ \forall p \in P \tag{23}$$

We also have to impose that the amount of pallets sent from i of the product p does not exceed the amount of pallets of p that i has:

$$\sum_{j \in J} \mathbf{y_{i,j,p}} \leq N_{i,p}, \ \forall i \in I, \ \forall p \in P \tag{24}$$

Lastly, we distribute the pallets of p sent from i to j in different transport means:

$$\sum_{k \in K} \mathbf{x_{i,j,p,k}} = \mathbf{y_{i,j,p}}, \ \forall i \in I, \ \forall j \in J, \ \forall p \in P \tag{25}$$

3.6 Lowerbound the shipments

On one hand, there is a limited amount of pallets that can fit in a certain vehicle, which is given by the constraint:

$$\sum_{j_2 \in J, p \in P} \frac{\mathbf{b_{i,j_1,j_2}} \cdot \mathbf{x_{i,j_2,p,k}}}{P_k} \leq \mathbf{s_{i,j_1,k}}, \ \forall i \in I, \ \forall j_1 \in J, \ \forall k \in K$$

Moving the term on the right hand side to the left side dividing, and P_k to the right side multiplying, the constraint will indicate that the (average) number of pallets sent per shipment in a certain transport mean k from i to its cluster centered at j_1 cannot exceed the maximum number of pallets that fit in k, which is P_k.

As we have done before, this constraint can be linearized using a max function:

$$\sum_{j_2 \in J, p \in P} \frac{\max(0, \mathbf{x_{i,j_2,p,k}} - M \cdot (1 - \mathbf{b_{i,j_1,j_2}}))}{P_k} \leq \mathbf{s_{i,j_1,k}}, \ \forall i \in I, \ \forall j_1 \in J, \ \forall k \in K$$

Thus, we need to define:

$$\mathbf{\bar{p}_{i,j_1,j_2,p,k}} = \mathbf{x_{i,j_2,p,k}} - M \cdot (1 - \mathbf{b_{i,j_1,j_2}}), \ \forall i \in I, \ \forall j_1, j_2 \in J, \ \forall p \in P, \ \forall k \in K \tag{26}$$

$$\mathbf{p_{i,j_1,j_2,p,k}} = \max(0, \mathbf{\bar{p}_{i,j_1,j_2,p,k}}), \ \forall i \in I, \ \forall j_1, j_2 \in J, \ \forall p \in P, \ \forall k \in K \tag{27}$$

And the constraint that we wanted will be:

$$\sum_{j_2 \in J, p \in P} \frac{\mathbf{p_{i,j_1,j_2,p,k}}}{P_k} \leq \mathbf{s_{i,j_1,k}}, \ \forall i \in I, \ \forall j_1 \in J, \ \forall k \in K \tag{28}$$

On the other hand, there is also a limited amount of weight load that a transport mean can handle, which is given by the inequality:

$$\sum_{j_2 \in J, \ p \in P} \frac{\mathbf{b_{i,j_1,j_2}} \cdot \mathbf{x_{i,j_2,p,k}} \cdot \sigma_p}{W_k} \leq \mathbf{s_{i,j_1,k}}, \ \forall i \in I, \ \forall j_1 \in J, \ \forall k \in K$$

This constraint can be linearized by adding the following ones:

$$\mathbf{\bar{w}_{i,j_1,j_2,p,k}} = \mathbf{x_{i,j_2,p,k}} \cdot \sigma_p - M \cdot (1 - \mathbf{b_{i,j_1,j_2}}), \ \forall i \in I, \ \forall j_1, j_2 \in J, \ \forall p \in P, \ \forall k \in K \tag{29}$$

$$\mathbf{w_{i,j_1,j_2,p,k}} = \max(0, \mathbf{\bar{w}_{i,j_1,j_2,p,k}}), \ \forall i \in I, \ \forall j_1, j_2 \in J, \ \forall p \in P, \ \forall k \in K \tag{30}$$

$$\sum_{j_2 \in J, p \in P} \frac{\mathbf{w_{i,j_1,j_2,p,k}}}{W_k} \leq \mathbf{s_{i,j_1,k}}, \ \forall i \in I, \ \forall j_1 \in J, \ \forall k \in K \tag{31}$$

Moving W_k to the right hand side and $\mathbf{s_{i,j_1,k}}$ to the left hand side in constraint (31), the left side will represent the weight sent per shipment from i to its cluster centered at j_1 in a certain transport mean k, and that value must not exceed the maximum weight permitted on a vehicle k, which is W_k.

3.7 Upperbound the shipments

To upperbound the shipments we add the next two constraints:

$$\mathbf{s_{i,j,k}} \leq M \cdot \delta_{i,j,k}, \ \forall i \in I, \ \forall j \in J, \ \forall k \in K \tag{32}$$

$$\sum_{j \in J, \ k \in K} \mathbf{s_{i,j,k}} \leq \bar{N}_i, \ \forall i \in I \tag{33}$$

With constraint (32) we are not letting the origin i send to its cluster centered at the destination j with the mean k if k cannot be used between them. This might happen, for example, if a lane was closed. Note that we need the constant M to be big enough.

With constraint (33) we state that there might be an upper bound to the amount of shipments that a warehouse can make in this time-period.

3.8 MOV constraints

The Minimum Order Value (MOV) is the minimum amount that customers are required to spend on the commodities that they ordered before those commodities can be shipped. While having a lower MOV implies that more shipments can be made, increasing the MOV will be responsible for performing less shipments.

After solving our problem we could decide which Minimum Order Value should be imposed on future orders: we could pick the minimum value of the orders that were made, or, since there might be outliers in that matter, we could pick the first quartile or the mean. Since the MOV is related to the cost of a shipment we could set a MOV for each transport mean. The MOV of a transport mean could be the mean value of the cost of the shipments performed with that mean in our solution. Anyways, the Minimum Order Value will be something that could be imposed in the future to prevent losing money in the shipments, and it should be based on the current and past orders received. Regardless, when setting a Minimum Order Value, at least a little forecast on how the immediate future will be for the company should be done.

In our case, instead of imposing a MOV on the orders, since we have the commitment of fullfilling every order, we could impose that all shipments with a certain transport mean have at least a certain profitability. This way, the shipments that generate less profit that the one specified will not be made. The value Σ_k of this profitability wanted on each shipment with the mean k will affect the feasibility of the problem: having a very large one might make this problem infeasible.

The next constraint states exactly what we have just discussed: the profitability obtained on all shipments from i to its group centered at j_1 with the mean k (profit obtained on the products transported minus the direct costs of the shipments) have to be at least the profitability wanted on shipments with k.

$$\sum_{p \in P} \mathbf{b_{i,j_1,j_2}} \cdot \mathbf{x_{i,j_2,p,k}} \cdot (\mu_p - \bar{\mu}_p) \geq \Sigma_k + \mathbf{sc_{i,j_1,k}} + \mathbf{lc_{i,j_1,k}}, \ \forall i \in I, \ \forall j_1 \in J, \forall k \in K$$

The left hand side term represents the profitability gained on each product sent from i to j_1 with k (computed as the product sell price minus the product acquisition cost), and the right hand side represents the profitability wanted in shipments with k plus the transportation costs generated by the shipments from i to j_1 with k. It can be linearized using the variable $\mathbf{p_{i,j_1,j_2,p,k}}$ defined in (27):

$$\sum_{j_2 \in J, p \in P} \mathbf{p_{i,j_1,j_2,p,k}} \cdot (\mu_p - \bar{\mu}_p) \geq \Sigma_k + \mathbf{sc_{i,j_1,k}} + \mathbf{lc_{i,j_1,k}},\ \forall i \in I,\ \forall j_1 \in J, \forall k \in K \tag{34}$$

3.9 Time constraints

Due to contract requirements there might be a lower bound of the number of times that a shop wants to be served every week:

$$\boldsymbol{\tau_{j_2}} \cdot \bar{\nu} = \sum_{i \in I, j_1 \in J, k \in K} \mathbf{s_{i,j_1,k}} \cdot \mathbf{b_{i,j_1,j_2}},\ \forall j_2 \in J$$

With this equality we are defining the delivery frequency of the shop j_2 as the number of times a week that j_2 receives a shipment, we have to take into account all the times that the groups which j_2 belongs to get sent commodities. We linearize it by doing:

$$\boldsymbol{\tau_{j_2}} \cdot \bar{\nu} = \sum_{i \in I, j_1 \in J, k \in K} \boldsymbol{\omega_{i,j_1,j_2,k}},\ \forall j_2 \in J \tag{35}$$

Then, we impose that the actual delivery frequency has to be greater or equal to the contractual delivery frequency:

$$\bar{\tau}_j \leq \boldsymbol{\tau_j},\ \forall j \in J \tag{36}$$

We could also have a limited amount of vehicles that we are able to use in the time-period:

$$\sum_{i \in I, j \in J} \mathbf{s_{i,j,k}} \cdot \kappa_{i,j,k} \leq \nu \cdot R_k,\ \forall k \in K \tag{37}$$

This constraint states that there are not being used more vehicles than the ones that can actually be used. The left side is the sum of the number of days that the vehicles are travelling, and the right side is the sum of the number of days that the vehicles are available to be travelling.

4. Formulation analysis

4.1 Problem size

Constraints involving maximums and minimums are actually constraints involving absolute values, and these are not differentiable constraints. Therefore, constraints involving maximums or minimums cannot be part of linear or quadratic programming problems. We can adapt our problematic constraints in an easy way to linear ones. For example, we could replace $a = \max(b, c)$ with $a \geq b$ and $a \geq c$. We could also replace $a = \min(b, c)$ with $a \leq b$ and $a \leq c$. After doing that, since all the constraints in our formulation will be linear, the objective function is linear, and there are integer and continuous variables, the presented formulation will be a Mixed-Integer Linear Programming (MILP) formulation of the Cost to Serve problem. A Mathematical Programming solver will be able to solve this MILP formulation using Branch and Cut, Gomory Planes, and other algorithms.

To do an asymptotic analysis we will choose $i := |I|$ to be the number of origins, $j := |J|$ the number of destinations, $k := |K|$ the number of transport means, and $p = |P|$ the number of products.

The number of variables of the formulation is $4ij^2pk + 2ij^2k + ijpk + ij^2 + 15ijk + ijp + ij + j + 2$.

The number of constraints in the formulation is:

$$4ij^2pk + 2ij^2k + ij^2 + 18ijk + ijp + 2ij + ip + jp + 2i + 2j + k + 2$$

which grows as $\mathcal{O}(ij^2pk)$.

4.2 Convergence analysis

We used Python to code the formulation and we decided to choose Gurobi and GLPK from Pyomo as solvers. On the one hand, Gurobi is one of the best Mathematical Optimization softwares, and we have been able to access it through a free academic license. On the other hand, Pyomo is an open-source collection of Python software packages for formulating optimization models, and GLPK is one of the solvers that it supports (among others such as AMPL or CPLEX).

When Gurobi is fed a problem, it states if it is infeasible. If the problem happens to be feasible, it returns the best feasible solution that it found along with a relative gap $r = |UB - LB|/|UB|$ (UB is an upper bound of the optimal solution and LB is a lower bound of it) that gives us an upper bound on how close the solution found is, percentwise, from the global minimum. It also prints out its progress: for each solution found Gurobi prints its value, the updated lower and upper bounds, the current gap, the time taken by the optimization process since it started, and other metrics.

When Pyomo is fed a model it does not print out the optimization progress. It just shows, when the optimization process is finished, whether the problem is feasible or infeasible. In the feasible case it also shows the optimal value of the objective function and it is possible to retrieve the soution.

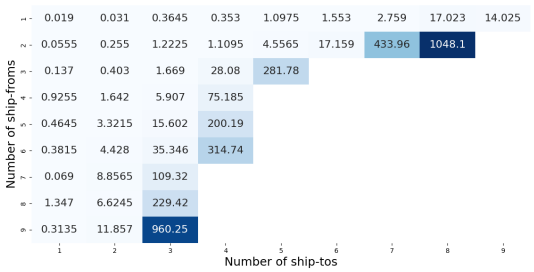
For the results we present throughout this section we used Gurobi v10.0.0 and Pyomo v6.4.3 with an Intel Core i7-8550U CPU @ 1.80 GHz.

After running different instances of the problem, we noticed that Pyomo usually took considerably longer to solve the problem than Gurobi. Since Pyomo does not print out messages to facilitate keeping

track of the optimization process, we chose to analyze the convergence time of the formulation for different instances of the Cost to Serve problem with Gurobi.

The following tables contain the average optimization time taken in seconds of different executions of the program with data randomly generated for each execution (we randomly chose the parameters while making sure that the whole set could be similar to a real-life case). We kept the number of products and the number of transport means fixed to check the dependency on the number of ship-froms and ship-tos. The horizontal and vertical axis are the number of ship-tos and the number of ship-froms respectively. For each number of origins and for each number of destinations we run multiple times the optimization program, we took the mean execution time taken and stored it in a table. In the tables a higher color intensity indicates a higher execution time.

(a) 1 product and 4 transport means



(b) 3 products and 4 transport means

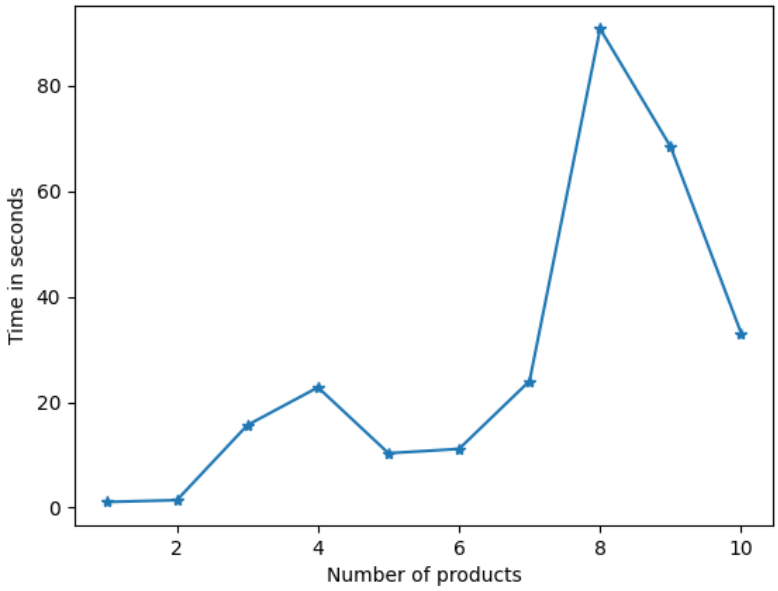
For figure 4a we set to have just one product and four different transport means, while for figure 4b we chose to have three products and four transport means. On the second figure the empty spaces indicate that the time taken was larger than 2400 seconds (20 minutes).

For the first table we can see that, since there is just one product, the execution time is very low. The number of ship-froms does not affect much to the execution time, which passes the threshold of 1 second for 5 ship-tos. However, for a higher number of ship-tos we might observe that the central part tends to have a higher execution time.

For the second table, since not all warehouses might have every product available, the complexity of the problem is larger than with just one product. We can see that having more destinations, or more origins, increases the time taken on the executions, which follows the expected behaviour. In this figure, in contrast with the other one, the locations on the table where the execution starts taking way more time than near cases to the left and above, might draw a stair (or even a diagonal line with a little imagination) that separates simple instances from more complex ones.

This way, we can see that for very simple instances of the problem, where the solution is almost straightforward, the increment on time execution happens when the number of destinations is augmented and not when the number of origins is. This is in line with the fact that the number of constraints of the problem grows as $\mathcal{O}(ij^2pk)$, so the the number of destinations has a larger effect on the number of constraints than the number of origins has, and, therefore, in the execution time of the formulation. However, for not so trivial instances, the duration of the execution depends more on both parameters.

If we fix instead the number of origins and destinations, we can compute the average execution time taken for different number of products if we run the program multiple times with different parameters on each execution. For two warehouses and five destinations the evolution of the time taken is:



We observe that for one and two products the execution is practically instantaneous. After that, the execution time grows and stays without much change from three to seven products. When we consider eight products the execution time skyrockets and then it decreases. One explanation would be that for little number of products the solution of the problem is liable to high variability (because it depends a lot on the locations of the warehouses and shops, i.e. it is very problem dependant), and when the number of products increases it stabilizes.

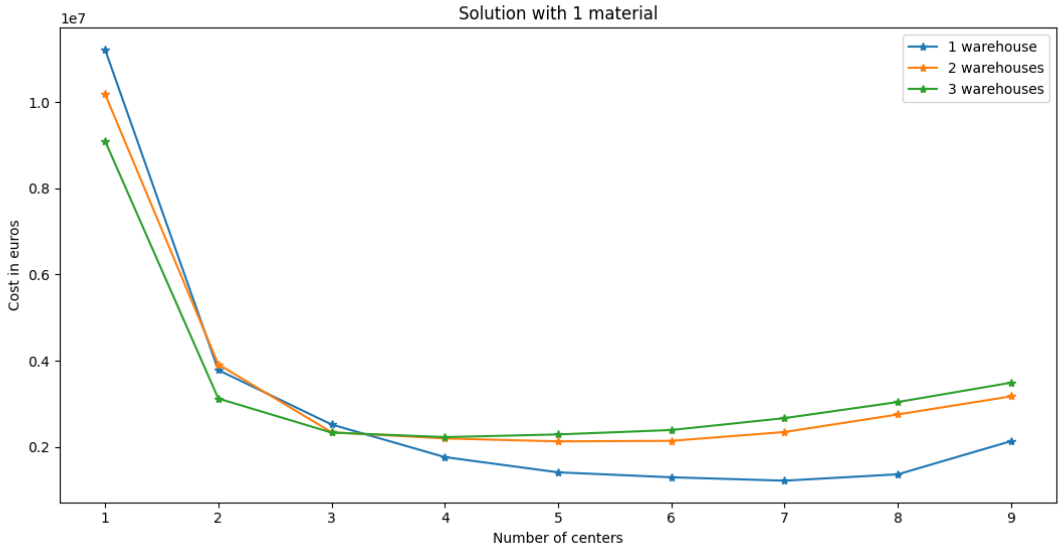
4.3 Hyper-parameter tuning

While the hyper-parameters c_1 and c_2 do not affect much, it must be mentioned that the hyper-parameter Φ_i, which can be found in constraint (4) and determines the number of clusters that will be formed for warehouse i, has a critical impact on the problem. Deciding a value for it highly impacts the execution time of the optimization process and the cost of the final solution. Thus, we have to choose an adequate number of clusters for every instance of the problem. It is not easy to infer the optimal number of clusters that should be built for every ship-from with small instances of the problem, and it is fairly difficult and sometimes even inviable to determine the optimal value of Φ_i for big instances.

One way of determining a value for Φ_i on small instances of the problem is considering $\Phi = \Phi_i \ \forall i \in I$ and using a grid search approach. Since the time needed for solving the problem is not very large, what should be done is executing the program for different values of Φ and choosing the one that gives the solution with less cost.

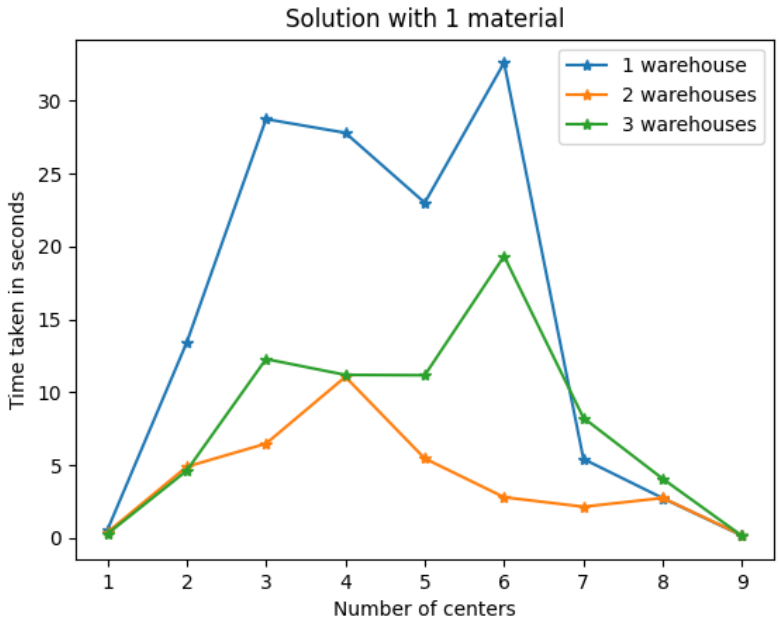
We will consider three cases in an example. Consider an instance where there is only one product, there are nine destinations, and we might have one, two or three warehouses. Since there are nine destinations, we can make up to nine clusters for each warehouse. We execute the problem with $\Phi = n, \ n = 1...9$ for the cases with one, two and three warehouses.

The following figure shows the cost of the optimal solution found for each number of ship-froms and for each number of centers chosen.



For one warehouse the optimal number of clusters would be $\Phi = 7$, while for two warehouses would be $\Phi = 5$, and for three warehouses would be $\Phi = 4$. Note that the cost with $\Phi = 1$ is the cost where all shops are grouped together, and the cost with $\Phi = 9$ is the cost without multi-drop. We can clearly see that choosing a value for Φ is not straightforward at all.

We can also take a look at how much time was taken to find the optimal solution for each case:



For one warehouse the optimal number of clusters ($\Phi = 7$) takes one of the lowest times to converge, and it is even six times faster than using one cluster less. For two warehouses the time taken with 5 clusters is the third highest, although all times are very similar. For three warehouses the time taken with 4 clusters is very similar to the ones taken with 3 and 5 clusters and their cost are close too.

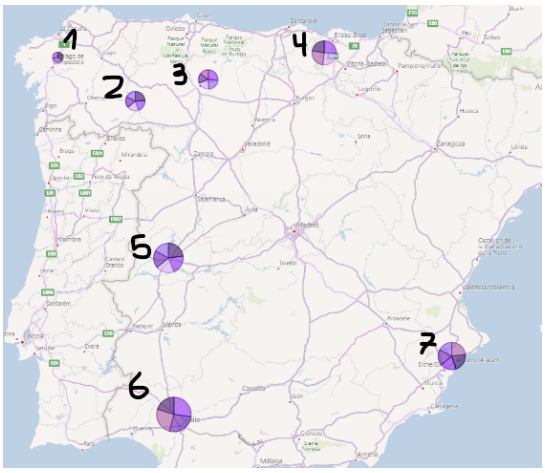
Increasing the complexity of the problem (in our example was adding warehouses) makes the costs and times taken with near values of Φ be close. This suggests that we could try to find a good enough value for Φ by finding the solution for some values of Φ that do not cover its whole domain, and using the number of clusters that yielded the lowest cost as the number of clusters to build on our problem.

5. A particular solution

We will study the optimal solution of an instance of the problem to see how the formulation works.

5.1 A particular solution

We generated dummy data in order to be able to study the optimal solution of the formulation on a particular instance of the Cost to Serve problem. We chose to study a problem with 2 warehouses and 7 shops, all of them located in Spain and selected randomly. The lanes that connect the warehouses and the shops have also been selected randomly, without taking into account if they exist or not.



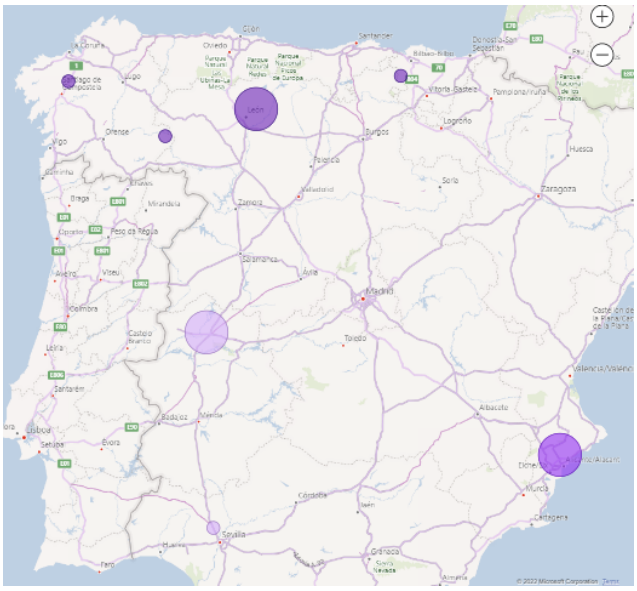
(a) Shops and the amount of each product requested

(b) Warehouses and the amount of products available

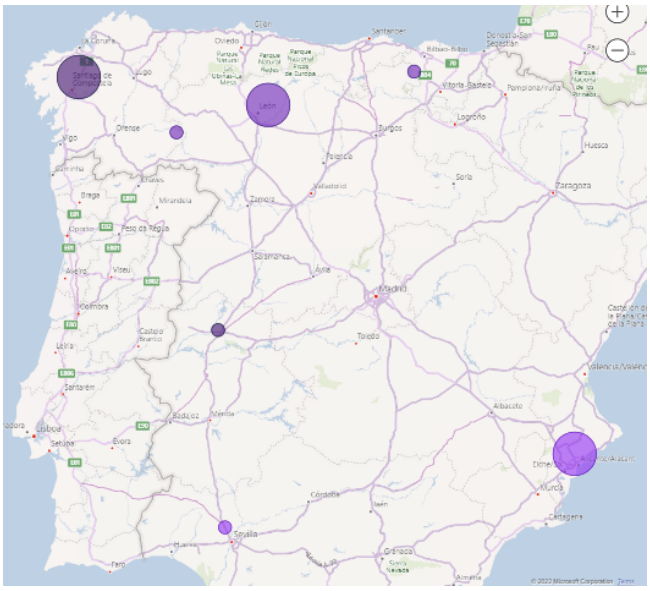
The numbering of the selected shops and warehouses used in these figures and in the ones throughout the section corresponds to the following postal codes:

Numbering	Postal code
Shop 1	15689
Shop 2	32330
Shop 3	24153
Shop 4	01477
Shop 5	10671
Shop 6	41219
Shop 7	03115
Shop 8	23688
Warehouse 1	24377
Warehouse 2	17810
Warehouse 3	23588

Table 1: Mapping between the numbering used and the postal codes



(a) Clusters for warehouse 1



(b) Clusters for warehouse 2

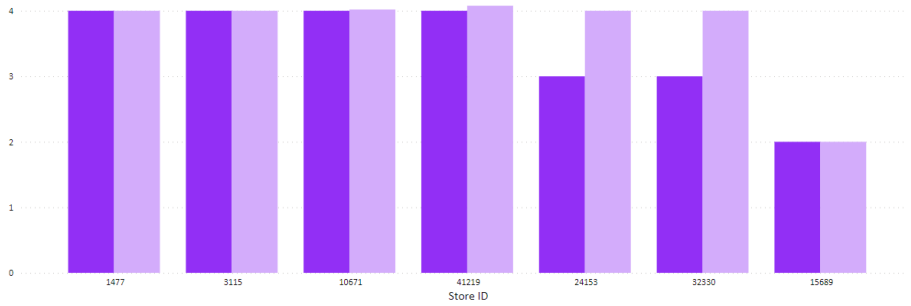


Figure 7: Contractual and Optimal Restricted delivery frequencies

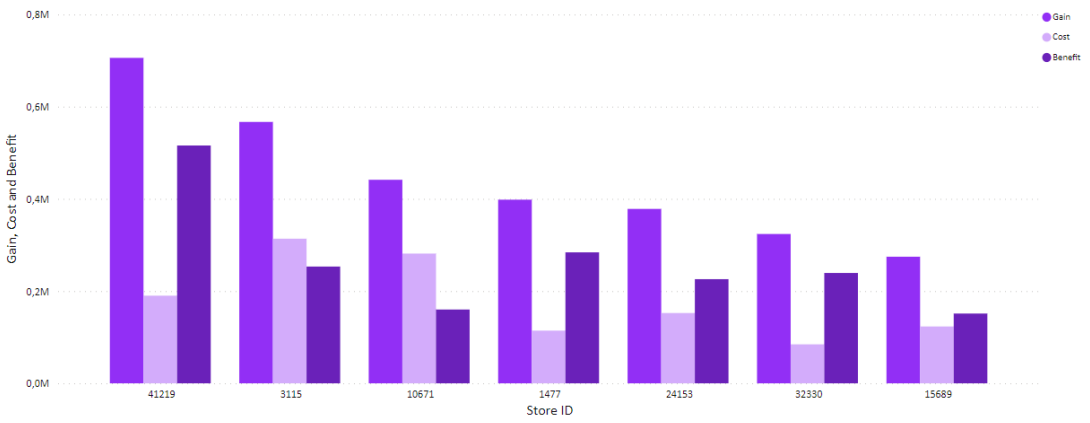


Figure 8: Gain, cost and benefit for each shop

Warehouse ID	Store ID	Vehicle Type	Shipments	Logistic Costs	Shipping Costs
24377	10671	truck	203	312.113,11	85540
24377	3115	truck	193	239.583,06	33920
24377	24153	truck	185	322.046,00	18320
17810	24153	ship	19	102.101,95	1900
17810	3115	ship	15	79.700,95	6720
17810	15689	ship	6	32.892,40	1110
24377	24153	ship	4	20.814,73	2719
Total			**625**	**1.109.252,20**	**150229**

Figure 9: Optimal shipments to be performed

In figures 6a and 6b, the big bubbles indicate that a shop (located in that postal code) is a center of a cluster for a warehouse, and the small bubbles indicate that a shop is not a center. Bubbles with the same colours belong to the same cluster. With that, these figures tell us that for warehouse 1: the shops 1, 2, 3 and 4 form the first group; the second group is formed by the shops 5 and 6; and shop 7 is left alone in the third group. For warehouse 2 we have built different clusters: the shops 2, 3 and 4 form the first group; the shops 1 and 5 belong to the second one; and the shops 6 and 7 form the third and last group.

Figure 7 shows the difference between the contractual delivery frequencies (parameters) and the optimal restricted ones (variables) for each ship-to. We can see that each shop is always delivered as many or more times a week as it requested.

Figure 8 shows the gain, cost anf benefit of each shop. We can see that all shops except 5 and 7 have a net profit higher than the cost. Additionally, every shop produces more gain than its cost, so every one is profitable.

Figure 9 shows which shipments have to be done, how many, and their cost. We can see that the shops 3 and 7 get served with two different transport means.

Finally, we get that the optimum value of the objective function (i.e. the total cost) is 4.62 million monetary units.

Analyzing these results we can see that the cluster network changed for the different ship-froms, which is great for minimizing the total cost, and that the delivery frequencies imposed are too harsh. Maybe if they were not imposed we could have gotten a better solution, that is, one whose transportation and emission costs are lower.

5.2 Improving this solution

After analyzing the results of 5.1 we might ask ourselves what would happen if there were not contractual delivery frequencies imposed, we would expect a great decline on the cost, since they were too restrictive. That is, if this instance of the Cost to Serve problem was solved with the formulation presented but without constraint (36).

The results of the optimization turn out to be very good. First, the execution time is reduced drastically: from an hour to 20 minutes. Having constraint (36) was a challenge for the solver to find the optimum quickly, because the value of the parameters $\bar{\tau}_j$ was unrealistic. And second, the value of the objective function (the cost) is now reduced to 2.05 million, which is half the previous cost.

The most notable changes on the solution follow:

Warehouse ID	Store ID	Vehicle Type	Shipments	Logistic Costs	Shipping Costs
17810	3115	ship	20	108.576,29	8416
17810	24153	ship	10	53.737,87	2889
24377	24153	ship	9	46.182,46	3575
17810	15689	ship	4	21.928,27	400
24377	24153	truck	4	3.480,74	400
24377	3115	truck	0	0,00	0
24377	10671	truck	0	0,00	0
Total			**47**	**233.905,63**	**15680**

Figure 10: Optimal shipments to be performed without constraint (36)

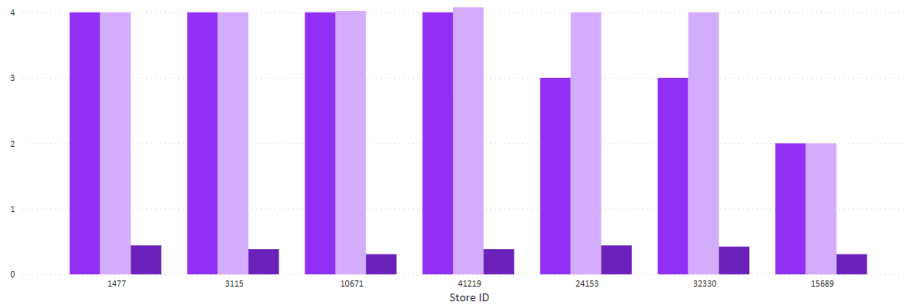


Figure 11: Delivery frequencies comparison: Contractual, Optimal Restricted, and Optimal Unrestricted

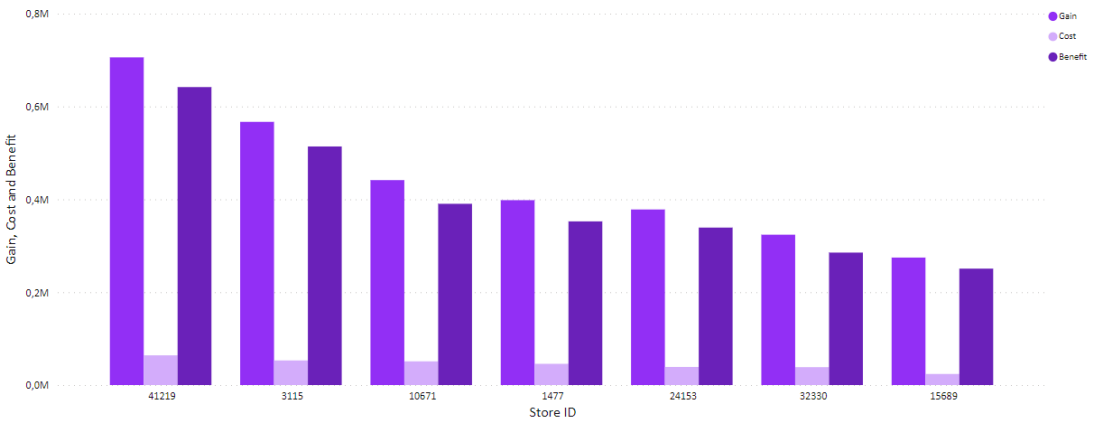


Figure 12: Gain, cost and benefit of each ship-to

The cluster network did not change much, but the number of shipments was reduced 13 times, which is the reason for the great reducement of the cost. We also see that the optimal delivery frequencies are around 0.5 times a week, which is considerably less than the ones obtained in the restricted problem. On top of that, the cost associated to each destination has declined significantly. Therefore, the goal for the

hypothetical company that tries to solve this Cost to Serve problem should be to try to reduce as much as possible the contractual delivery frequencies in order to reduce the total cost.

5.3 New scenarios

Due to globalization, constant change is present on every aspect of Today's World. For that reason, our formulation should be tested against small changes on the particularities of our problem. Little changes on product demand will not affect our solution (see section 6), but other things might do: we could be losing a client or gaining one, we might lose some capabilities, or we might expand and be able to satisfy a greater product demand. Because of that, we have to study the resilience of our network and the stability of our solution in the sense of how small perturbations to our problem affect the solution.

The three following scenarios are based on the particular improved instance (without constraint (36)) of the Cost to Serve problem that we are studying. Each one is different, but all of them focus on studying how the network changes for similar instances to our initial improved problem 5.2.

5.3.1 Removing a warehouse

Removing a warehouse will simulate the effect of losing capabilities or deciding that having a certain warehouse is not profitable enough to mantain it. To do that, we executed the program without warehouse 2. The cost of the objective function dropped from 2.05 million to 1.88 million, and the clusters of warehouse 1 did not change (they are the same as in figure 6a). However, an increment of the stock capacity of warehouse 1 was required.

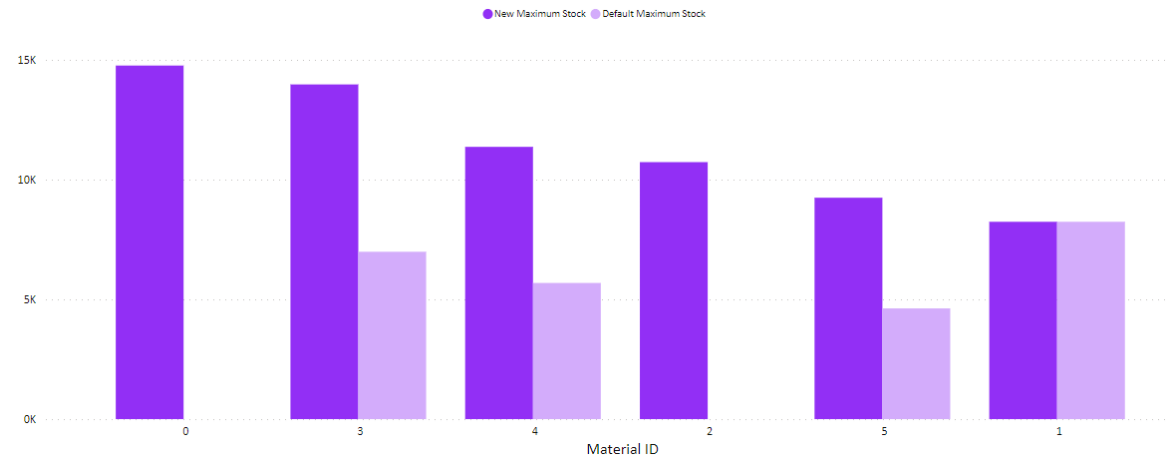


Figure 13: New maximum stock and default stock of warehouse 1

Even though the cost of just having warehouse 1 is lower than having both ship-froms, this high increment on the capacity of warehouse 1 might not be profitable enough to remove a warehouse voluntarily.

5.3.2 Adding a warehouse

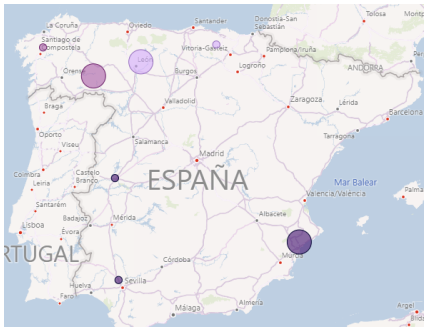
Adding a warehouse simulates the effect of expanding and being able to cover more market share. Since the origins that we have are both in the north of Spain, it would be a good idea to expand to the south and acquire a warehouse in that area.

We selected a random postal code in the south of Spain and generated random dummy data to locate a warehouse in it. We then ran the formulation with the same parameters as in section 5.2.

Figure 14: Warehouses and the amount of products available

The optimal cost in this scenario is 1.07 million, almost half of the cost in section 5.2. Consequently, expanding by acquiring a ship-to in a key location would be a priority in this instance of the Cost to Serve problem.

The clusters to build in the optimal solution are now the following:



(a) Clusters for warehouse 1

(b) Clusters for warehouse 2

(c) Clusters for warehouse 3

A comparison between the groups formed in this section and the groups formed in 5.1 has to be done. For warehouse 1 major changes on the clusters have been done. The first group has been split into two: shops 1 and 2 form the first group and shops 3 and 4 belong to the second one, while the destinations 5, 6 and 7 have been grouped together. For warehouse 2 there has been little change: while shop 6 has changed from group 3 to group 2, all the other ship-tos have stayed in the group that they used to belong.

Finally, warehouse 3 grouped shops 1, 2, 3, 4 and 7 into the first group, and let the shops 5 and 6 be alone in groups 2 and 3 respectively.

On the following figure we look at how many deliveries are now made to each shop.

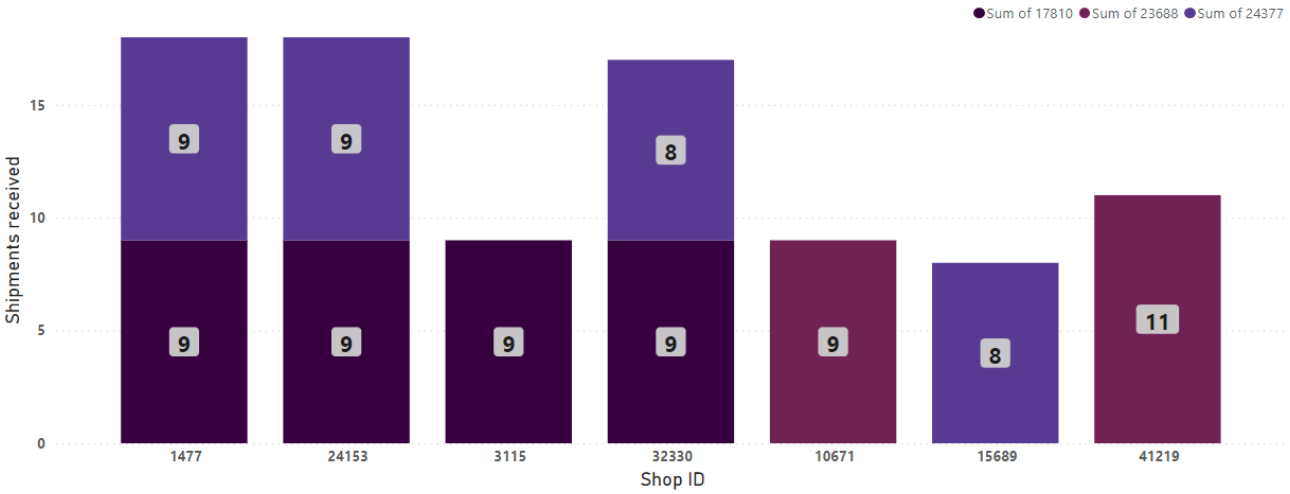


Figure 16: Deliveries received by each shop. See table 1 for codes.

We can see that warehouse 1 does not send to its third group (shops 5, 6 and 7), warehouse 2 does not send to its second group (shops 1, 5 and 6), and warehouse 3 does not send to its first group (shops 1, 2, 3, 4 and 7). With this, the optimal solution is telling us that it is a good practice to have a cluster for each origin reserved for the destinations that will not be receiving any products from the warehouse. Therefore, one should have this in mind when choosing a value for $\Phi = \Phi_i \ \forall i \in I$.

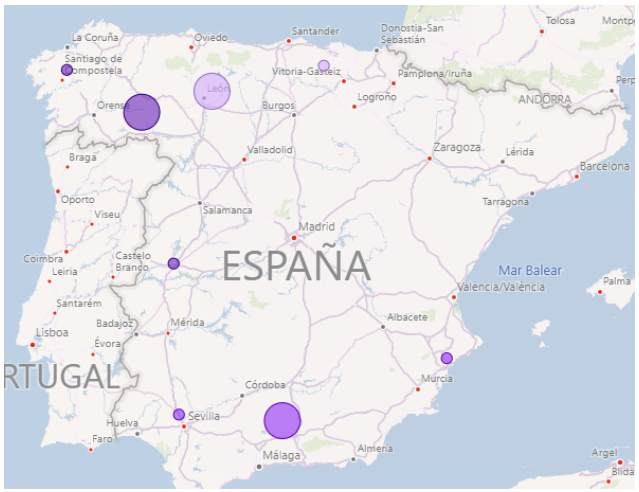
Another thing to notice is that while the ship-tos in the north (1, 2, 3 and 4) tend to be sent shipments together, the ship-tos in the south (5, 6 and 7) tend to be sent individually. Thus, the use of the multi-drop technique (constraints (1) to (4)) and choosing a good value for the parameters Φ_i can really improve the way that logistic companies serve by grouping near enough destinations and not grouping far enough ones.

5.3.3 Adding a shop

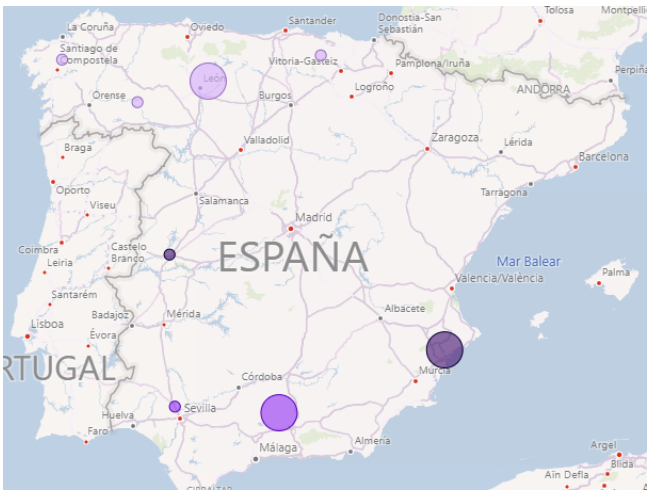
Adding a shop simulates the effect of increasing market share. To do that, we chose a random location in Spain to locate shop 8 and we generated a random order for it. We then updated the parameters and ran the formulation.

The execution time was very similar to the one in 5.2, but its cost went up from 2.05 million to 2.44 million. That increment is obvious because our network of warehouses did not improve (we stayed with two and at the same locations), but the demand increased. At least a comparison of the benefits and costs of the two cases should be done to decide whether to accept the orders of the new destination or not. Vehicle availability should also be considered, and not having enough stock product will make our problem infeasible.

Another thing to look at in order to make that decision is to study how the cluster network changes. Changing the groups a lot will imply a cost, since new routes will have to be computed and established.



(a) Clusters for warehouse 1



(b) Clusters for warehouse 2

We can see that by adding shop 8 (between shops 6 and 7) the clusters do change a lot. For warehouse 1 the clusters change: shops 1, 2 and 5 form the first group; shops 3 and 4 form the second one; and shops 6, 7 and 8 belong to the third cluster. For warehouse 2 the clusters change less: shop 1 now belongs to the first group, shop 7 now belongs to the second one, and the new shop is assigned to the third group with shop 6. This change in the cluster network is a result of not having a good location of the warehouses to satisfy the desired demand.

6. Stability analysis

Little changes on product demand can be modelled as small perturbations on the parameters. The cases that we examined in section 5 can also be modelled as perturbations on the parameters: solving an instance of the Cost to Serve problem without a certain ship-from would be equivalent to solving the initial problem with that origin and changing the parameters so that all that are related to the specified warehouse are null or cause that it is not used. The same would happen for losing a destination. Since the relation between solving a problem P and the same problem where a ship-from or a ship-to has been added is the same as the relation between a problem $\tilde{P}$ and the same problem where a ship-from or a ship-to has been removed, adding an origin or a destination can also be modelled as a perturbation of the parameters of an initial problem.

To study how small perturbations on the parameters affect the solution to our formulation, we will study the stability of a linear programming problem. We will base the study in Stephen M. Robinson's [9].

First of all, any pair of dual linear programming problems is written in standard form as:

$$\begin{aligned} \text{minimize} \quad & \langle c, x \rangle \\ \text{subject to} \quad & Ax \leq b, \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \text{maximize} \quad & \langle u, b \rangle \\ \text{subject to} \quad & A^T u \leq c \\ & u \leq 0 \end{aligned}$$

The goal is to prove that a necessary and sufficient condition for the primal and dual sets of a solvable and finite-dimensional linear programming problem to be stable under small but arbitrary perturbations in the parameters of the problem is that both of these sets are bounded. We will conclude that the distance from any pair of solutions of the perturbed problem to the solution sets of the original problem is then bounded by a constant multiplied by the norm of the perturbations. Therefore, if we assume that our Cost to Serve formulation has bounded primal and dual solution sets —which is a reasonable assumption in real-world instances—, the optimal solution (or the set of optimal solutions) of our problem will be stable.

Clearly, any linear programming problem can be written in primal and dual forms as:

$$\begin{aligned} \text{minimize} \quad & \langle c, x \rangle \\ \text{subject to} \quad & Ax - b \in \mathcal{Q}^* \\ & x \in \mathcal{P} \end{aligned} \qquad \text{(P)} \qquad\qquad \begin{aligned} \text{maximize} \quad & \langle u, b \rangle \\ \text{subject to} \quad & c - uA \in \mathcal{P}^* \\ & u \in \mathcal{Q} \end{aligned} \qquad \text{(D)}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are nonempty polyhedral convex cones in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively, and the asterisk denotes the dual cone:

$$\mathcal{P}^* = \{z \in \mathbb{R}^n : \langle z, x \rangle \geq 0 \ \forall x \in P\}$$
$$\mathcal{Q}^* = \{z \in \mathbb{R}^m : \langle z, u \rangle \geq 0 \ \forall u \in Q\}$$

It should also be noticed that our MILP formulation for the Cost to Serve problem can be written in these forms.

The objective is to establish the conditions that (P) and (D) must satisfy so that for small perturbations on the parameters A, b, c, the problems (P) and (D) remain solvable and their solution sets are stable. By

stable we mean that for any primal and dual solutions x' and u', the distances between x' and the solution set of (P) and between u' and the solution set of (D) are bounded by a constant multiplied by the size of the perturbations.

If we denote the interior of a set X as $\text{int}(X)$, we can define the following concepts.

Definition: the constraints of (P) are regular if $b \in \text{int}(A(\mathcal{P}) - \mathcal{Q}^*) = \text{int}(\{Ap - q^* : \forall p \in \mathcal{P}, \forall q^* \in \mathcal{Q}^*\})$.

Definition: the constraints of (D) are regular if $c \in \text{int}((\mathcal{Q})A + \mathcal{P}^*) = \text{int}(\{qA + p^* : \forall q \in \mathcal{Q}, \forall p^* \in \mathcal{P}^*\})$.

Defintion: if the constraints are not regular, we call them singular.

For example, if we take $\mathcal{P} = \mathbb{R}^n$ and $\mathcal{Q} = \mathbb{R}^m$, the system $Ax - b \in \mathcal{Q}^*$ and $x \in \mathcal{P}$ is simply the system of linear equations given by $Ax = b$ and $x \in \mathbb{R}^n$, since $\mathcal{Q}^* = \{0\} \subset \mathbb{R}^m$. That means that b belongs to the interior of the range of A, which is true for any $b \in \mathbb{R}^m$ if and only if A has full row rank (if A is a squared matrix it has to be non-singular). Thus, the regularity of the constraints might be thought of as a natural generalization of the concept of full row rank to more general systems involving inequalities and constrained variables. We must observe that for $A(\mathcal{P}) - \mathcal{Q}^*$ to have an interior it is not necessary that either $A(\mathcal{P})$ or $\mathcal{Q}^*$ have one.

For example, if we consider the system with constraints:

$$\begin{array}{rcl} x_1 & & \leq 1 \\ & x_2 & \leq 1 \\ x_1 & -x_2 & = 0 \\ x_1, & x_2 & \geq 0 \end{array}$$

and we define:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

we can write them as $Ax - b \in \mathcal{Q}^*, \; x \in \mathcal{P}$. We have that $\mathcal{P} = \mathbb{R}^2_+$, so $\mathcal{P}^* = \mathbb{R}^2_+$, and we also have that $Ax - b = \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \\ x_1 - x_2 \end{bmatrix} \begin{matrix} \leq 0 \\ \leq 0 \\ = 0 \end{matrix}$, so $\mathcal{Q}^* = \mathbb{R}^2_- \times \{0\}$ and then $\mathcal{Q} = \mathbb{R}^2_- \times \mathbb{R}$. The set $A(\mathcal{P}) = \mathbb{R}^2_+ \times \mathbb{R}$ has an interior, $\mathcal{Q}^* = \mathbb{R}^2_- \times \{0\}$ does not, and $A(\mathcal{P}) - \mathcal{Q}^* = \mathbb{R}^2_+ \times \mathbb{R}$ does and b belongs to it.

This system cannot be written as a system of pure inequalities depending on the non-negative variables x_1, x_2, that is, as a $k \times 2$ system of the form $Dx \leq d, \; x \geq 0$ while satisfying the requirement $d \in \text{int}(D(\mathbb{R}^2_+) + \mathbb{R}^k_+)$. It is essential to retain the capability to handle equations and unconstrained variables if the regularity conditions are to be satisfied. The use of $\mathcal{P}$ and $\mathcal{Q}$ eliminates the necessity for distinguishing between inequalities and equations, and between non-negative and unconstrained variables. Therefore, this use of sets greatly simplifies the notation on the problem.

We now state three lemmas that will help prove the result that ensures the stability of the solutions of our Cost to Serve formulation.

Lemma 1: let K and L be two nonempty polyhedral convex cones in $\mathbb{R}^k$ and $\mathbb{R}^l$ respectively, let M be a $k \times l$ matrix, and let $m \in \mathbb{R}^k$. Suppose that the set $F = \{x \in L : Mx - m \in K\} \subset \mathbb{R}^l$ is not empty.

Then, there exists some $\theta \geq 0$ such that for each $x \in L$, $d(x, F) \leq \theta \cdot d(Mx - m, K)$, where for a point y and a set A, $d(y, A) = \inf\{||y - a|| : a \in A\}$.

Proof: can be deduced from the fundamental theorem of Hoffman [7] on approximate solutions of systems of linear inequalities.

The second lemma is a technical one that examines the behaviour of the solution set F when M and m are slightly perturbed. We note B as the unit ball in $\mathbb{R}^l$, and we note for a closed set $A \subset \mathbb{R}^l$ and $\epsilon \geq 0$. $A + \epsilon B = \{x \in \mathbb{R}^l : d(x, A) \leq \epsilon\}$.

Lemma 2: let K, L, M, m, θ, F be as in lemma 1. Suppose that there exists $\mu \geq 0$ such that $F \subset \mu B$. Then, $\forall \epsilon > 0$ and $\forall M', m'$ with $\delta := \max\{||M' - M||, ||m' - m||\} < \theta^{-1}\epsilon/(1 + \epsilon)$, the set $F' = \{x \in L : M'x - m' \in K\}$ satisfies that $F' \subset F + \lambda_\epsilon \delta B$ with $\lambda_\epsilon = (1 + \epsilon)(1 + \mu)\theta$. If $\theta = 0$ we interpret θ^{-1} as $+\infty$.

Proof: choose $\epsilon > 0$ and pick M' and m' as described. If $F' = \emptyset$ the result is obviously true. Suppose that $F' \neq \emptyset$ and let x' be any point in $F' \subset L$. Let x_0 be the closest point in F to x'. By lemma 1:

$$||x' - x_0|| = d(x', F) \leq \theta \cdot d(Mx' - m, K) \tag{i}$$

Since $M'x' - m' \in K$, we have that $d(Mx' - m, K) \leq d(Mx' - m, M'x' - m') = ||(Mx' - m) - (M'x' - m')|| \leq ||M' - M|| \cdot ||x'|| + ||m' - m|| \leq \delta(1 + ||x'||)$, so:

$$d(Mx' - m, K) \leq \delta(1 + ||x'||) \tag{ii}$$

We also have that $||x'|| = ||x_0 - x_0 + x'|| \leq ||x_0|| + ||x_0 - x'|| \leq \mu + ||x' - x_0||$. Substituting in (ii):

$$d(Mx' - m, K) \leq \delta(1 + \mu + ||x' - x_0||) = \delta(1 + \mu) + \delta||x' - x_0|| \tag{iii}$$

Substituting (iii) in (i):

$$||x' - x_0|| \leq \theta \cdot d(Mx' - m, K) \leq \theta \cdot \delta(1 + \mu) + \theta \cdot \delta||x' - x_0|| \tag{iv}$$

Rearranging terms:

$$||x' - x_0||(1 - \theta\delta) \leq \theta\delta(1 + \mu) \tag{v}$$

Since $1 - \theta\delta > 1 - \theta\frac{\theta^{-1}\epsilon}{1+\epsilon} = \frac{1}{1+\epsilon} > 0$, we can conclude that $||x' - x_0|| \leq \frac{\theta\delta(1+\mu)}{1-\theta\delta} \leq (1 + \epsilon)(1 + \mu)\theta\delta$.

The third lemma shows that regular linear systems are precisely those that remain solvable under all sufficiently small perturbations in the parameters.

Lemma 3: the linear system $Ax - b \in \mathcal{Q}^*,\ x \in \mathcal{P}$ is regular if and only if there exists some $\eta > 0$ such that for any A', b' with $\max\{||A' - A||, ||b' - b||\} < \eta$, the system $A'x - b' \in \mathcal{Q}^*,\ x \in \mathcal{P}$ is solvable.

Proof: Let's see $\Longleftarrow$: if the system is singular, that is, if $b \notin \text{int}(A(\mathcal{P}) - \mathcal{Q}^*)$, then there exist points b' arbitrarily close to b such that $b' \notin A(\mathcal{P}) - \mathcal{Q}^*$. That means that the system $Ax - b' \in \mathcal{Q}^*, x \in \mathcal{P}$ is not solvable.

Let's see $\Longrightarrow$: suppose that there $\nexists \eta > 0$ with the previous property. We want to see that $b \notin \text{int}(A(\mathcal{P}) - \mathcal{Q}^*)$. By assumption, we can find sequences $\{A_n\}_{n\in\mathbb{Z}}$ and $\{b_n\}_{n\in\mathbb{Z}}$ that converge to A and b respectively, such that $\forall n \in \mathbb{Z}$, the system

$$A_n x - b_n \in \mathbb{Q}^*,\ x \in \mathcal{P}$$

has no solution.

By a variant of the Farkas lemma (derived from theorem 3.5 of [4]), this system is not solvable if and only if there exists some $w_n \in \mathbb{R}^m$ such that:

$$w_n A_n \in -\mathcal{P}^*,\ w_n \in \mathcal{Q},\ \langle w_n, b_n \rangle > 0$$

Since w_n is not zero, we can suppose that $||w_n|| = 1$ without loss of generality. Then, the sequence $\{w_n\}$ is contained in the unit sphere, which is a compact set, so it has a convergent subsequence $\{w_m\} \underset{m\to\infty}{\longrightarrow} w$ (with $||w|| = 1$).

Taking the limit in the previous system and using that $\mathcal{P}^*$ and $\mathcal{Q}$ are closed, we have that:

$$wA \in -\mathcal{P}^*,\ w \in \mathcal{Q},\ \langle w, b \rangle \geq 0$$

Now consider any point $y = Ap - q^* \in A(\mathcal{P}) - \mathcal{Q}^*$, with $p \in \mathcal{P}$ and $q^* \in \mathcal{Q}^*$. We have that:

$$wA \in -\mathcal{P}^* = \{z \in \mathbb{R}^n : \langle z, x \rangle \leq 0\ \forall x \in P\} \implies \langle wA, p \rangle \leq 0 \tag{a}$$

$$q^* \in \mathcal{Q}^* = \{z \in \mathbb{R}^m : \langle z, u \rangle \geq 0\ \forall u \in \mathcal{Q}\} \implies \langle w, q^* \rangle \geq 0 \tag{b}$$

Using (a) and (b):

$$\langle w, y \rangle = \langle w, Ap - q^* \rangle = \langle wA, p \rangle - \langle w, q^* \rangle \leq 0 \tag{vi}$$

Then, $\langle w, y \rangle \leq 0 \leq \langle w, b \rangle$ and, since $w \neq 0$, it follows that the hyperplane $\{v \in \mathbb{R}^m : \langle w, v \rangle = 0\}$ separates b from the convex cone $A(\mathcal{P}) - \mathcal{Q}^*$. Therefore, b cannot be in the interior of the cone $A(\mathcal{P}) - \mathcal{Q}^*$.

Theorem: the folowing are equivalent:

1. The constraints of (P) and (D) are regular.
2. The sets of optimal solutions of (P) and (D) are nonempty and bounded.
3. There exists an $\epsilon_0 > 0$ such that for any A', b', c' with $\epsilon = \max\{||A' - A||, ||b' - b||, ||c' - c||\} < \epsilon_0$, the two dual problems

$$\begin{array}{cc} (P') & \\ \text{minimize} & \langle c', x \rangle \\ \text{subject to} & A'x - b' \in \mathcal{Q}^* \\ & x \in \mathcal{P} \end{array} \qquad \begin{array}{cc} (D') & \\ \text{maximize} & \langle b', u \rangle \\ \text{subject to} & c' - uA' \in \mathcal{P}^* \\ & u \in \mathcal{Q} \end{array}$$

   are solvable.

If these conditions are satisfied, then there exist constants $\epsilon_1 \in (0, \epsilon_0]$ and γ such that for any A', b', c' with $\epsilon < \epsilon_1$, any x' solving (P') and any u' solving (D'), we have that $d((x', u'), S_P \times S_D) \leq \gamma \cdot \epsilon$, where S_P and S_D are the sets of optimal solutions for (P) and (D) respectively.

Proof: If (P) and (D) are not both solvable, then by the duality theorem of linear programming ([4] theorem 4.6), at least one must be infeasible, thus 1 and 2 are both false. To prove their equivalence for the case in which (P) and (D) are both solvable, it suffices to prove that the set of solutions of (D) is bounded if and only if $b \in \text{int}(A(\mathcal{P}) - \mathcal{Q}^*)$. The corresponding result for the solution set of (P) follows by symmetry. We first note that the perturbated function $f(y) = \inf_x \langle c, x \rangle : Ax - y \in \mathcal{Q}^*, x \in \mathcal{P}$ associated with (P) is a proper convex function since (P) and (D) are solvable, and that the effective domain of f is $A(\mathcal{P}) - \mathcal{Q}^*$. By theorem 23.4 of [10] the set of subgradients $\partial f(b)$ is nonempty and bounded if and only if $b \in \text{int}(A(\mathcal{P}) - \mathcal{Q}^*)$. However, it is well known that the set of solutions of (D) is precisely $\partial f(b)$, and this establishes the equivalence of 1 and 2.

The equivalence of 1 and 3 is shown by applying lemma 3 and then noting that the feasibility of both (P') and (D') is equivalent, by the duality theorem, to their solvability.

To see the second part of the theorem, suppose that the three equivalent conditions in the statement of the theorem hold. We note that the set of points (x, u) solving (P) and (D) is the solution set of the linear system:

$$\begin{bmatrix} 0 & -A^T \\ A & 0 \\ c & -b^T \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} - \begin{bmatrix} -c^T \\ b \\ 0 \end{bmatrix} \in \begin{bmatrix} \mathcal{P}^* \\ \mathcal{Q}^* \\ \{0\} \end{bmatrix},$$
$$\begin{bmatrix} x \\ u \end{bmatrix} \in \begin{bmatrix} \mathcal{P} \\ \mathcal{Q} \end{bmatrix}$$

We also note that this set is bounded by hypothesis. As the set of points (x', u') solving (P') and (D') is the solution set of the system obtained from the previous one by replacing A, b, c by A', b', c'. Using lemma 2 we complete the proof.

This theorem ensures the stability of the optimal solution of a linear programming problem with respect to small perturbations on the parameters if the sets of optimal solutions of the primal and dual sets are bounded. Thus, if we assume that our Cost to Serve formulation has bounded primal and dual solution sets, the optimal solution of our problem will be stable under small perturbations on the parameters, such as changes in product demand, or the addition or substraction of certain ship-froms or ship-tos (if the facility network is big enough).

7. Conclusions and further research

In this research we provided a Mixed-Integer Linear Programming formulation for the Cost to Serve problem where we considered the costs to be transportation costs and environmental costs due to the emissions produced by the transportations. We used the multi-drop technique to reduce the costs by giving the possibility of combining shipments of near enough destinations, and we considered the possibility of using different transportation means to carry out the shipments. We also considered other real-life constraints that might exist.

We analyzed how stable is the solution of our problem, in the sense of how small perturbations of the parameters affect the solution of the problem. We concluded that the distance from the optimal solution of a perturbed instance of the problem to the set of optimal solutions of the original instance is bounded by a constant multiplied by the size of the perturbations. These perturbations may be changes in product demand or changes in the facility network.

However, this MILP formulation can not be used with very big instances of the Cost to Serve problem, since the number of variables grows fast. For that reason, it would be a good idea to implement a Genetic Algorithm or another metaheuristic to find a good solution and use it as a start solution to solve our formulation with a Mathematical Optimization solver. That way we could find an optimal solution much faster than by just using a Mathematical Optimization solver.

References

[1] Freight transport demand by mode - EU27. https://www.eea.europa.eu/data-and-maps/figures/freight-transport-demand-by-mode-eu27. Accessed: December 26th 2022.

[2] Global CO_2 emissions from fossil fuels. https://ourworldindata.org/co2-emissions. Accessed: December 26th 2022.

[3] Specific CO_2 emissions per passenger-km and per mode of transport in Europe. https://www.eea.europa.eu/data-and-maps/daviz/specific-co2-emissions-per-passenger-3#tab-chart_1. Accessed: December 26th 2022.

[4] Adi Ben-Israel. Linear equations and inequalities on finite dimensional, real or complex, vector spaces: A unified theory. *Journal of Mathematical Analysis and Applications*, 27(2):367–389, 1969.

[5] Alan Braithwaite and Edouard Samakh. The cost-to-serve method. *The International Journal of Logistics Management*, 9(1):69–84, 1998.

[6] Reza Zanjirani Farahani and Mahsa Elahipanah. A genetic algorithm to optimize the total cost and service level for just-in-time distribution in a supply chain. *International Journal of Production Economics*, 111(2):229–243, 2008.

[7] Alan J Hoffman. On approximate solutions of systems of linear inequalities. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 174–176. World Scientific, 2003.

[8] Pranavi Pathakota, Kunwar Zaid, Anulekha Dhara, Hardik Meisheri, Shaun D Souza, Dheeraj Shah, and Harshad Khadilkar. Learning to minimize cost-to-serve for multi-node multi-product order fulfilment in electronic commerce. *arXiv preprint arXiv:2112.08736*, 2021.

[9] Stephen M Robinson. A characterization of stability in linear programming. *Operations Research*, 25(3):435–447, 1977.

[10] R Tyrrell Rockafellar. *Convex analysis*, volume 18. Princeton university press, 1970.

[11] AC Williams. Marginal values in linear programming. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):82–94, 1963.