

A Territory-based Approach for the Facility Assignment Problem with a Minimum Cost Flow Formulation

Matheus Suknaic

Loggi Ltda

matheussuknaic@gmail.com

Fillipe Goulart

Loggi Ltda

fillipe.gsm@tutanota.com

Juan Camilo

Loggi Ltda

juankmilofg@gmail.com

ABSTRACT

This document proposes a solution process for the Online Facility Assignment problem applied to Logistics. Given a number of Expedition Centers capable of delivering packages to the final customers, the goal is to devise a policy for assigning each package to a center in order to reduce the total distance between them while respecting possible volume and exclusivity regions. Also, the assignment must happen in an online manner. Here we propose a two-phase approach. First, we construct demand regions for each EC via a mixed linear programming formulation. Afterwards they are employed for the online package assignment. The algorithm is further improved with a minimum cost flow version. We show how the network flow approach can improve a lot on the processing time while keeping the same quality on objective value. Finally, a full walk-through illustrates the applicability of the method in a real-world scenario.

KEYWORDS. Online Facility Problem, Minimum Cost Flow, Logistics.

1. Introduction

The supply chain consists in the entire system of producing and delivering a package or a service. The transportation of these packages from a buyer to a seller is an important element dealt by Logistic companies, and the proper optimization of its inner steps can lead to significant costs savings [Watson et al., 2013].

A logistic chain is normally composed of three portions, also called “miles”. The first mile is responsible for the pick-up of packages from the sellers and marks their entering in the company’s logistic network. During the middle mile the products travel through the network’s hubs and warehouses, and this is an important step when dealing with large distances. Finally, the packages move to a final warehouse – which we will refer to as an Expedition Center (EC), preferably closer to its final destination –, from which it follows the last mile delivery to the customer. Each of these steps has a number of sub-steps, each one with their peculiarities and specific room for improvement. In this work, we deal with the last step right before the last mile one: the problem of selecting a proper EC from which the product should be dispatched.

Suppose there is a hub that should process packages for a city, and it has two ECs at disposal for that. The decision should be performed in such a way that we minimize the last mile delivery costs while respecting possible constraints, such as maximum/minimum volume of products and priority zones. A simple and popular rule of thumb to follow is to assign each package to the closest EC (with respect to a given distance function). Thus, with the knowledge of all products to be processed, there are mathematical approaches to help us with this task and still respect the constraints.

However, in large scale logistic companies we tend to lack the luxury of aggregating all incoming packages first and processing them later. In such cases, there may be time windows for the packages or for some vehicles, or maybe there is just not enough physical space in the hub to accommodate the volume. Hence, we need a policy to process a package and assign it to an Expedition Center as soon as it becomes available.

This dynamic nature complicates the operation, as exemplified in Figure 1. Assume, for simplicity, that a hub is connected to two final ECs, each one capable of handling at most three packages. It receives six packages in a given day with the order shown. If we adopt a greedy approach that attributes each package to the closest EC in the order they arrive, we get the results of the top right panel. Despite reducing the last mile total distance, it violates the green EC capacity. On the other hand, a similar policy that “blocks” an EC as soon as it fills its capacity results in the middle right panel with respected constraints but higher transportation costs since the last (farther) package was assigned to the orange EC. The arguably best scenario here is the bottom right one, which respects the capacities with smallest total distance.

1.1. Related work

The problem discussed in this work is closely related to the *Online Facility Assignment*, which is a variant of the well-known Facility Location [Laporte et al., 2019] but with the dynamic availability of the customers. It is also a generalization of the Online Weighted Matching [Kalyanasundaram e Pruhs, 1993], in which the facilities correspond to the right side of the bipartite graph and have a capacity larger than one.

Two works that handle directly this problem are [Ahmed et al., 2020] and [Muttakee et al., 2020]. They deal with the Online Facility Assignment when the facilities are either positioned on a straight line or on unweighted graphs. The authors develop and analyze approximate algorithms with interesting upper bounds.

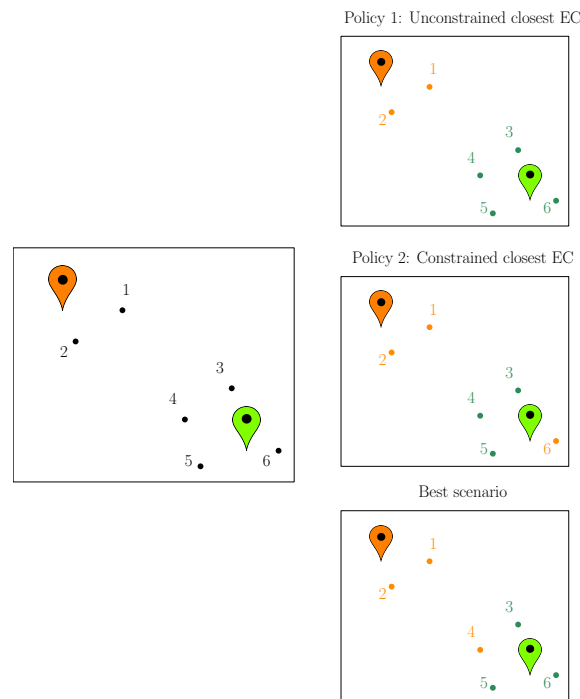


Figure 1: Two common policies for assigning an incoming package to an EC.

Despite the evident advances in the field, the assumptions in the previous works represent limitations when applying these algorithms in real world logistics scenarios. More specifically, in practice (i) we require the use of street distances rather than an analytical metric or norm; (ii) the facility capacities can be different, while the cited works can only deal with the same number; and (iii) a totally online solution has no guarantees of recurrence, i.e., the Expedition Centers may be assigned to different subregions each day. In practice, companies may perform partnerships with drivers, so a degree of predictability is desired.

In spite of not being able to predict incoming packages in a proper order, most large scale logistics companies enjoy some kind of regularity in the deliveries, i.e., the package distribution in a city or similar region remains relatively stable over time.¹ Thus, we can make use of historical information to build specific delivery zones for each EC such that the incoming new volume can be better controlled. These zones, which we will call *demand regions*, can be constructed in such a way that the total last mile distance from the EC to the packages inside of it is minimized while the operational constraints are respected. Moreover, since these are historical data, we can dispense with the dynamic issues and employ a mathematical approach with full knowledge of the packages to be processed.

This work is divided as follows. After this first section with the problem's motivation and a quick overview in the literature, the second section models the demand regions' construction via a Mixed Integer Linear Programming (MILP) problem. The third section proposes a Minimum Cost Flow (MCF) modeling to improve on processing time while keeping similar levels of optimality gap. The fourth section discusses the obtained results, and the final one concludes the work.

¹The actual *volume* may fluctuate more, but as shown later the proposed approach can also work when we have an upper bound in the forecast.

2. A Territory-based approach for the Facility Assignment Problem

The main premise behind the proposed method revolves around recurrence of the delivery distribution in a specific region. Assume in a given city we extract historical packages in a typical day as shown by the left panel of Figure 2. As we can see, most of the load is concentrated around the green EC. Suppose this accounts for about 50% of the total volume. If both centers have a similar capacity and we would like to balance the load, we could construct a smaller demand region around the green EC, and another covering the rest of the whole territory as shown in the right panel of the same figure. We can interpret these demand regions such that, assuming the package distribution does not change too much, about 50% of the new packages will fall in the green demand region, even if we cannot predict exactly where or when. With such an approach, the policy of processing incoming packages in a hub becomes as simple as deciding for membership in a demand region.

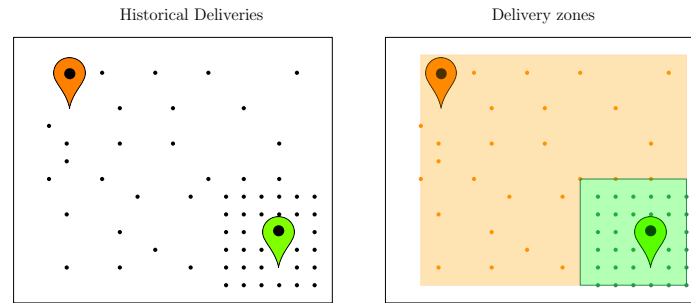


Figure 2: Demand regions construction attempting to get at most 50% of the load to the green EC.

2.1. Planning phase: Constructing Demand Regions

Since the previous example was simple, it was possible build the regions manually. However, a real world scenario would have tens of thousands of packages in a day, multiple Expedition Centers, each one with minimum/maximum volumes and priority zones.

To develop a mathematical formulation of the problem, let N be the number of Expedition Centers in a given city or similar territory and M the number of sample packages in a typical day, which we will refer to as a *training set*. Let $x_{i,j}$ be a binary variable with value 1 if the i -th EC is assigned to the j -th package. The first step is to decide which ECs should deliver which packages, and this is provided in the formulation below.

$$\text{minimize} \quad \sum_{i=1}^N \sum_{j=1}^M \text{dist}_{i,j} x_{i,j} \quad (1)$$

$$\text{subject to} \quad \sum_{i=1}^N x_{i,j} = 1, \quad \forall j = 1, \dots, M \quad (2)$$

$$\sum_{j=1}^M x_{i,j} \leq \text{max_volume}_i, \quad \forall i = 1, \dots, N \quad (3)$$

$$\sum_{j=1}^M x_{i,j} \geq \text{min_volume}_i, \quad \forall i = 1, \dots, N \quad (4)$$

$$x_{i,j} = 1 \text{ if } \text{coords}_j \in \text{multipolygon}_i, \quad \forall i = 1, \dots, N \quad (5)$$

$$\forall j = 1, \dots, M$$

$$x_{i,j} \in \{0, 1\}$$

in which

- Constraint (2) ensures each package is assigned to one EC;
- Constraints (3) and (4) safeguard the number of packages assigned to the i -th EC is within specific limits;²
- Constraint (5) provides that an EC will deliver in specific subregions. This may be useful specially in zones of risk, where a specific EC may be more prepared to handle it entirely;
- Finally, the packages should be assigned in such a way that the overall traveled distance is minimized as in Equation (1).

Notice that, with this formulation, the distance can be any general function and does not need to respect properties such as being a metric, triangular inequality etc. In fact, any useful “cost” function can be adopted instead.

This problem can be seen as a Transportation Problem with unbalanced sources, or a more general Assignment Formulation [Luenberger e Ye, 2008]. While there are efficient methods to handle the basic versions of these problems, the volume and polygon constraints requires a more general Mixed Integer Linear Programming (MILP) solver.

Once we know which packages are attributed to each EC, we need to actually *build* the demand regions. There needs to be some parsimony here, since the way these regions are described affects the next step when we need to determine to which region an incoming new package belongs. For simplicity, we choose to describe such demand region by the actual subset of packages assigned to it after problem (1) was solved.

2.1.1. Handling package volume variations

As explained in the previous section, the constructed demand regions are such that if the total volume of new incoming packages and their distribution are similar to the training data, then the package assignments will respect the volume constraints while attempting to minimize the total distance. While delivery distributions tend to be stable in large scale companies, the total volume may largely fluctuate over the time (maybe over months and sometimes in a single week).

Fortunately, the same approach is still adequate here with a slight adaptation. Suppose the training data in a typical day has 10k deliveries, and the ECs maximum capacities are 5k and 8k for the green and orange ECs, respectively. Assume known from forecasting information that the largest total volume in the city has a peak of 15k packages but with a similar distribution. If we wish to safeguard both centers to the peak value, we do not necessarily need to change the training dataset. Instead, here are two possibilities:

- Scale the volume constraints: since the expected peak volume is 1.5 times larger than the training data volume, the maximum volumes of the ECs should be $\approx 3,333$ and $\approx 5,333$ for the green and orange ECs, respectively;

²Notice that the default values would be $\text{min_volume} = 0$ and $\text{max_volume} \rightarrow \infty$, and if there is no priority polygon this is reduced to the simple case where each EC is assigned to the closest package.

- Scale the individual package demands: each point in formulation (1) represents a single package. Instead, we could set a demand of $d_j = 1.5, \forall j = 1, \dots, M$, and replace $x_{i,j}$ with $d_j x_{i,j}$ in equations (3) and (4).

Both approaches should return the same results, although the second is more flexible if the packages have different individual demands. The net effect in this example is to tell the green EC to handle only a third of the share of the city in peak conditions, so the orange demand region would cover a portion of the green square of Figure 2. We show a more detailed example in the results section.

2.2. Execution Phase: Processing new incoming packages at real time

The previous phase can be used at any moment to construct or update the demand regions of a territory based on its history. Once these regions are built, we can use them to process new incoming at a hub at real time by simply finding to which demand region it belongs.

Since the demand regions are composed of individual historical packages, given a new incoming package as shown in Figure 3, we could compute a regular Euclidean distance from each point, find the closest, and assign the new package to its corresponding EC. A possibly more robust approach which we prefer here is to employ a more general k nearest neighbors (k -NN) [Wikipedia contributors, 2022] method, which finds the k closest points and selects the group that appears the most. The right panel of the same figure shows an example with $k = 5$, a common default value in many numerical software and adopted here.

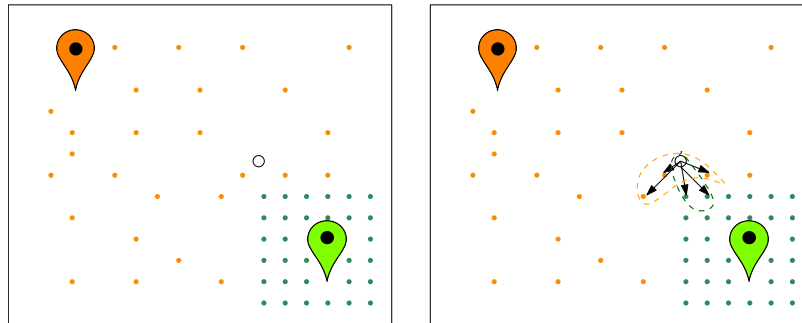


Figure 3: Execution Phase: use of k -NN to assign a new package to an Expedition Center. The new package is closer to three orange points and two green ones, so it is assigned to the orange EC.

3. A Minimum Cost Flow formulation for the Planning Phase

While formulation (1) is simple and robust enough, MILP algorithms are still exponential and may be demand high processing times for large problems. While this may be o.k. since the algorithm is meant to be executed offline, obtaining a faster execution can be interesting if it becomes a subproblem of a larger problem – such as the Facility Location – or maybe just to alleviate timeouts if implemented in a web application. Therefore, another possibility to model packages' assignment to expedition centers is to use a Minimum Cost Formulation (MCF), which can be solved by efficient polynomial algorithms [Magnanti et al., 1993].

To model our problem as a network flow, we define a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ the set of edges. Extending the previous notation, let $\mathcal{V}_p$ be the set of packages and $\mathcal{V}_{EC}$ the set of Expedition Centers. Then, the vertex set is the union of all packages and expedition centers plus a terminal node t , namely $\mathcal{V} = \mathcal{V}_p \cup \mathcal{V}_{EC} \cup \{t\}$.

To understand the edge set, consider Figure 4. The first layer contains edges from each package to an EC. If a package is inside an EC's exclusivity region, then it cannot have edges to the others EC. The second layer contains edges from each expedition center to the terminal node. To get a mathematical expression, let $\mathcal{E}^-(n)$ be the set of edges entering a node $n \in \mathcal{V}_{EC}$, i.e.,

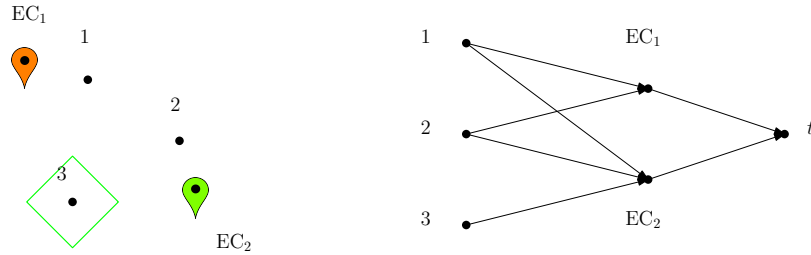


Figure 4: MCF formulation of the demand regions creation.

$$\mathcal{E}^-(n) = \begin{cases} \{(m, n)\} & \text{if } \text{coords}_m \in \text{multipolygon}_n \\ \{(m', n) | \forall m' \in \mathcal{V}_p\} & \text{otherwise} \end{cases}$$

Then, the first layer can be written as $\mathcal{A} = \{\mathcal{E}^-(n) | \forall n \in \mathcal{V}_{EC}\}$ and the second as $\mathcal{B} = \{(n, t) | \forall n \in \mathcal{V}_{EC}\}$, from which we get $\mathcal{E} = \mathcal{A} \cup \mathcal{B}$.

Lastly, each edge in $\mathcal{G}$ has a cost and capacity, attributed through a cost function $c : (v, w) \rightarrow \mathbb{Z}$ and maximum flow capacity function $u : (v, w) \rightarrow \mathbb{Z}$. Similarly, each vertex in $\mathcal{G}$ has a supply/demand, defined by a function $b : v \rightarrow \mathbb{Z}$. Generally if $b(v) > 0$ we call it a supply and if $b(v) < 0$, we call it a demand.

Once $\mathcal{G}$ is defined, we model our problem as a minimum cost flow problem:

$$\text{minimize} \quad \sum_{(v,w) \in \mathcal{E}} c(v, w) f(v, w) \quad (6)$$

$$\text{subject to} \quad \sum_{w \in \mathcal{V}} f(v, w) - \sum_{w \in \mathcal{V}} f(w, v) = b(v), \quad \forall v \in \mathcal{V} \quad (7)$$

$$\sum_{v \in \mathcal{V}} b(v) = 0 \quad (8)$$

$$0 \leq f(v, w) \leq u(v, w) \quad (9)$$

in which

- Constraint (7) ensures that difference between the flow that enters and leaves a node is exactly the supply/demand of that node.
- Constraint (8) states that the sum of all demands and supplies is zero, safeguarding that all packages will be delivered and that all expedition centers' minimum volume capacities will be respected.
- Constraints (9) guarantees that expedition centers' maximum volume capacities will be respected and that no negative flows are allowed.
- Finally, the packages should be assigned in such a way that the overall cost is minimized when considering the flows going through each edge (6).

4. Experimental Results

To assess the results of the proposed approach, we adopt the Loggi Benchmark for Urban Deliveries (LoggiBUD) dataset [Loggi, 2021]. Its goal is to provide a realistic³ set of instances for routing related problems. The instances represent large scale deliveries (ranging from thousands to tens of thousands each), their distribution follows socioeconomic parameters instead of artificial probability distributions, and we use street distances returned from an Open Source Routing Machine (OSRM) server [Luxen e Vetter, 2011] to better represent the geography of the territory. The repository is open source and free for academical and commercial use.

At the moment of writing we have datasets available from three cities in Brazil, and we pick Rio de Janeiro – the largest one – for our experiments. The data is divided into a `training` and a `dev` portions, each one with multiple instances representing deliveries in a day. Semantically speaking, the former portion indicates deliveries that already happened, so we have full information about them, while the latter represents the testing data.

The experimental part is divided into two sections. The first one compares the performance of the proposed MCF formulation (6) with the MILP one (1). The second simulates the planning and execution phases of the proposed method with two instances from the `training` and `dev` datasets and assess how well the demand regions help solving the Online Facility Assignment problem.

4.1. Comparing MCF and MILP formulations

For this section, we select a training instance with a total of 24,638 deliveries, which is a large value and typical of the reality of our company. The experimental setup consists of the following independent parameters:

- Solver: we used the COIN-OR Branch and cut solver for the MILP formulation and a Push-relabel algorithm for the MCF;
- Number of Expedition Centers: varied between 5, 10 and 15, which are common numbers. Their locations were randomly selected as packages of the training sample;
- Number of packages: varied between 25%, 50%, 75% and 100% of the training sample.

Each combination of these three variables was executed 27 times with varying values of minimum, maximum volumes and specific subregions. Each specific subregion was modeled through a randomly generated polygon inside the city boundaries.

The distances between each EC and the packages were computed via an OSRM local server, with data compiled for Brazil. To cope with the integrality assumptions, they were rounded to the nearest integer, which has already a good resolution since the values are returned in meters. Also and of course, each solver handled the exact same inputs. With that, we compared each scenario with respect to the objective value (total street distance) and total processing time (in seconds).⁴

Figure 5 compares the objective value and processing time between solvers. As can be observed and displayed by the left panel, the returned objective values are the same for each scenario (in fact, the pairwise differences between them averages to zero). However, with respect to the processing time, there seems to be a huge advantage for the MCF formulation.

³The data do not actually come from *real* customers. It is synthetically constructed from socio-economical sources such as IBGE, IPEA and OpenStreetMaps. However, it correlates very well with the deliveries we face at our company, Loggi. Thus, it can be considered realistic enough to represent real-world challenges.

⁴The proposed algorithm was implemented in Python 3.8 and all the experiments were run on a single thread on an Intel(R) Core(TM) i7-1165G7 @ 2.80GHz, with 16GB of RAM available and running Ubuntu 20.04.

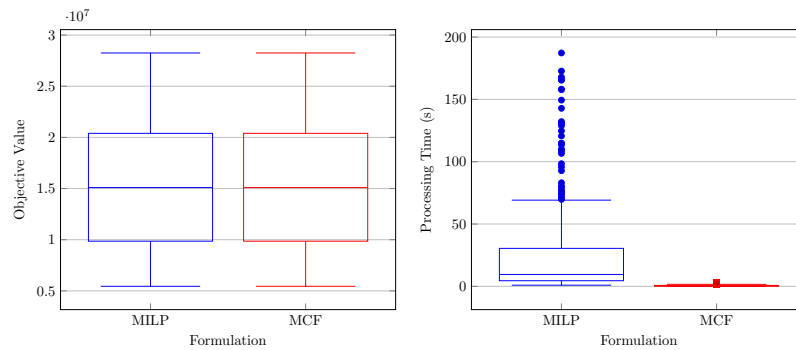


Figure 5: Comparison between the solvers with respect to objective value (left) and processing time (right).

Figure 6 goes into more detail when comparing the processing times for different numbers of ECs and packages. All three panels are at the same scale. Thus, we can see first that the processing times seems to be similar regardless of the number of ECs for both formulations, and they grow almost linearly with the number of packages. With that said, the MILP solver seems to demand much more time, in some cases ranging over 120 seconds. The MCF seems almost constant in comparison, but it has a similar behavior as elucidated in Figure 7. Even so, its overall processing time is much smaller, staying well below the two seconds mark.

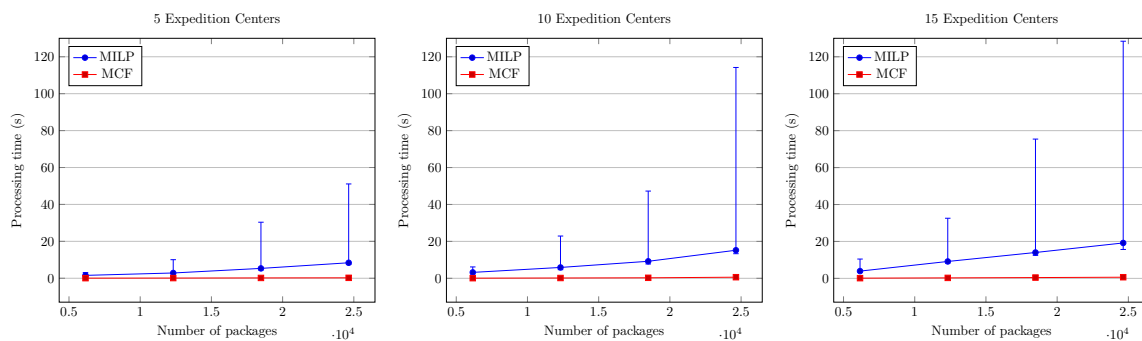


Figure 6: Processing times for each solver grouped by number of ECs. The vertical lines are limited by the first and third quartile, with the median as the center.

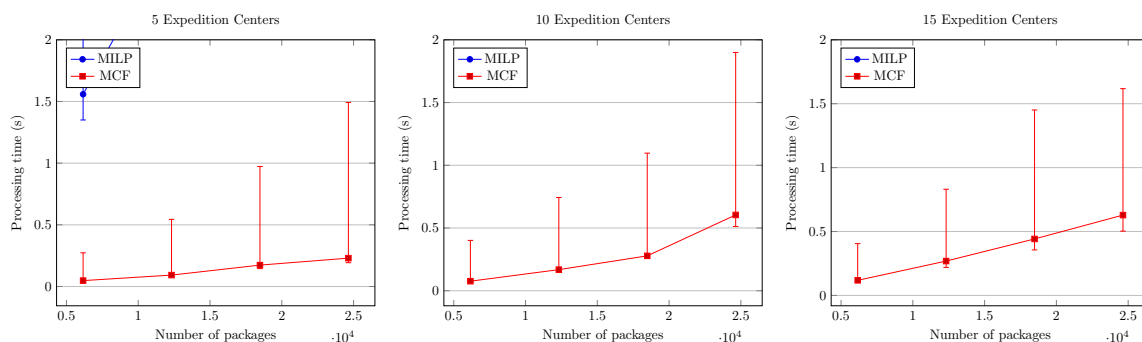


Figure 7: Same as Figure 6 but zoomed at the MCF values.

4.2. Assessing demand regions quality

This section shows how the proposed method would be employed in a real world scenario. Suppose there are three ECs at the city, and one (the black one) has an exclusive region at the island as shown in Figure 8. From the same training data as before, in a given day we had a total volume of 24,638 packages. Finally, say that each EC has a maximum capacity equal to 15k.

At first we could create demand regions such that, if the city volume is closer to the training data, the capacities would be satisfied. However, say that our analysts predict a peak volume of 35k. As explained in section 2.1.1, we can still safeguard us from this peak volume using the same training data.

Following the first approach, we construct demand regions using a scaled maximum volume of $15k \times 24,638 / 35k \approx 10,560$ packages each. The resulting regions are shown in the top panel of Figure 8. Notice how the exclusive region constraint is respected for the black EC, even though the blue one is closer to that territory. Also, the region nearby the blue EC is the most dense one, so, in order to respect the capacity constraints, the algorithm automatically shifted some portions of it to the black EC, which is the second closest one. Table 1 shows the distribution of packages for each EC in this training data.

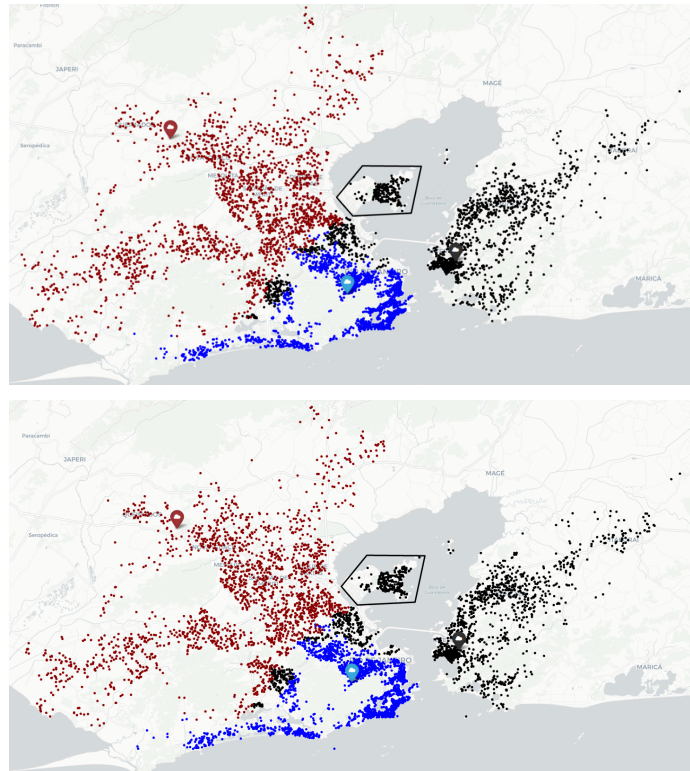


Figure 8: Top: demand regions from the training data. Bottom: Resulting package distribution with the test data.

With the demand regions constructed, we can emulate a typical day with a test data, and in this case we assign each package to a proper EC as explained in the Execution Phase (section 2.2). The results are also shown at the bottom of Figure 8 and in the last two columns of Table 1. As shown in the figure, the actual distribution is very close to the predicted one, and the percentage of packages assigned to each EC is very similar.

EC	Latitude	Longitude	Training data		Test data	
			Share (%)	Volume	Share (%)	Volume
Black	-22.897626	-43.096075	21.9%	5393	21.6%	7035
Blue	-22.940158	-43.251042	42.9%	10559	42.9%	13983
Brown	-22.734807	-43.505515	35.2%	8686	35.5%	11550

Table 1: Package distributions for planning phase in the training data and after the execution phase with the test data.

Even though the peak volume was of 32,568 packages, by re-scaling the ECs maximum volume, we were able to keep the volume below the original maximum volume of each center as shown in the last columns of Table 1. Of course, our conservative approach meant that the black EC had to deliver at farther zones in order to reduce the load in the blue one and prevent a breakdown in peak conditions. Therefore, the objective function was actually *worse* than what it could have been if we knew exactly the actual volume in that day, and with almost all problems we face a compromise between cost reduction and safety in the operation. This trade-off is observed more thoroughly in moments where there is a remarkable change in the number of packages to be assigned, such as the Black Friday or Christmas. For these scenarios it is fundamental to minimize the package assignment cost, however without overloading any of the expedition centers. With that said, our proposed approach is simple enough that it can be updated with a good frequency if required.

5. Conclusion

This work proposes an approach to solve the Online Facility Assignment problem applied in a real-world logistics setting. Right before the last mile step in a logistic company, there must be a decision about the best Expedition Center to deliver a package. This arrangement must be performed as soon as a package becomes available and must be such that the total distance to the ECs to their packages are minimized while possible volume and exclusivity zones are respected. Here, we take advantage of a recurrence property of most large scale companies and create *demand regions* indicating the delivery zones of each EC. Their construction is first formulated via a mixed integer linear programming (MILP), and later improved with a minimum cost flow (MCF) version.

We first show how the MCF formulation is able to employ more efficient algorithms that solve the problem with the same optimality quality but much faster processing times. Then, we show a walk-through of how the method can be employed in practice to prevent Expedition Centers from working too far from their capacities. While the algorithm depends on appropriate forecasting values to obtain good performances, it is still simple enough and can be updated as often as a company needs to follow package distributions over time.

For future works it is possible to focus more on the description of the demand regions. Keeping a large number of packages may place a good toll of some database, specially if we plan on adopting the method for multiple cities. Thus, we can work on either reducing their size or maybe adopting a different representation.

References

- Ahmed, A. R., Rahman, M. S., e Kobourov, S. (2020). Online facility assignment. *Theoretical Computer Science*, 806:455–467.
- Kalyanasundaram, B. e Pruhs, K. (1993). Online weighted matching. *Journal of Algorithms*, 14(3): 478–488.

- Laporte, G., Nickel, S., e Saldanha-da Gama, F. (2019). Introduction to location science. In *Location science*, p. 1–21. Springer.
- Loggi (2021). loggibud: Loggi benchmark for urban deliveries. <https://github.com/loggi/loggibud>.
- Luenberger, D. G. e Ye, Y. (2008). Linear and nonlinear programming.
- Luxen, D. e Vetter, C. (2011). Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, p. 513–516, New York, NY, USA. ACM. ISBN 978-1-4503-1031-4. URL <http://doi.acm.org/10.1145/2093973.2094062>.
- Magnanti, T., Ahuja, R., e Orlin, J. (1993). Network flows: theory, algorithms, and applications. *PrenticeHall, Upper Saddle River, NJ*.
- Muttakee, S. A., Ahmed, A. R., Rahman, M., et al. (2020). New results and bounds on online facility assignment problem. *arXiv preprint arXiv:2009.01446*.
- Watson, M., Lewis, S., Cacioppi, P., e Jayaraman, J. (2013). *Supply Chain Network Design: Applying optimization and analytics to the global supply chain*. Pearson Education.
- Wikipedia contributors (2022). K-nearest neighbors algorithm — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=1077256005. [Online; accessed 20-May-2022].

6. Acknowledgments

We would like to thank Loggi for the funding and assistance of this work. Also, we appreciate Gabriela Surita for proposing the LoggiBUD initiative and José Carlos Maldonado for the encouragement of this and many other researches. Finally, we are grateful for team colleagues, namely Angélica Custódio, Hélio Santana, Janaína Fernandes, Luan Leonardo, Mariana Pinheiro and Pedro Leite Rocha. Thank you all for the support.